"HOW CAN I..."

System Technical Guide

Pumps and motors management

Schneider Electric

Version 2 - 06 2008

Table of contents

Chapter	Topic	Page
1	Introduction	2
2	Selection phase	5
3	Design phase	17
4	Configuration phase	33
5	Implementation phase	43
6	Operation phase	55
7	Water application example	59
8	Glossary	77

1-Introduction

Purpose

The aim of this STG (system technical guide) is to provide recommendations, guidelines, and examples to help you develop the best application for pump and motor control.

The guide targets especially the medium size process control application setting up a mix of cost effective and advanced motor control systems. The three main values of these applications are:
- Ease of design
- Ease of maintenance
- Ease of extension

More complex applications using intelligent Motor Control Centers (iMCC) or MV motors are not targeted by this technical guide. In the same way compact pumping applications using ATV61 multi-pumps cards are not described below. Additional technical guides will be provided for these applications.

The guide is structured in terms of different phases of the application life cycle:

■ **Selection phase**
The guide proposes criteria to select the most appropriate solution for motor control and connection type. The aim is to build an automation architecture that integrates motor control solution and to propose guidelines that will improve the effectiveness of the design phase.

■ **Design phase**
A system architecture is described and used in all the following chapters. Hardware wiring diagram and software architecture flowchart are presented. At this stage, all components from SCADA system to starters are combined together to build a motor control application. Therefore, the data to be exchanged between components has to be clearly defined.

■ **Configuration phase**
Recommendations are provided to facilitate the configuration of the system using dedicated software.
The data exchanged between components are configured in the same way.

■ **Implementation phase**
Management of the various starters requires dedicated functions and objects in the PLC, SCADA and HMI applications. Several software packages are designed for these applications. And detail information about programming with these software is provided in this chapter..

■ **Operation phase**
An effective motor control application has to provide relevant diagnostic and process control features to help the operation of the plant. Motor control diagnostic and control information is listed in this chapter.

Preliminary

This System Technical Guide (STG) can be used for all projects that use pumps and motors. Water application projects are one of the most important ones due to the significant numbers of pumps, and the importance of pump function in the process. Therefore a wastewater application is developed at the end of the document to illustrate the recommendations given in the previous chapters.

The System Technical Guide does not intended to replace any specific product documentation. Readers are considered to have already known how to use the products described In this STG.

1-Introduction

Solution tested and validated

In order to ensure all information provided by this STG is tested and validated, a solution platform has been developed to validate and document all explanations provided in this guide. The platform integrates various motor control architectures with various types of communication. It comprises all components required to build a complete motor control application from SCADA system to starter devices.

2-Selection phase

Table of contents

Principle	6
Selecting starter mode	8
Selecting motor control devices	10
Selecting architecture	14

Principle

Introduction

This chapter presents the various steps required to select the most appropriate starter components as well as the automation architecture which will ensure control.

Selection criteria

Each process control project has specific requirements and constraints that influence the selection of a motor control solution.
The project specification present the characteristics that determine the selection criteria to be used in the selection steps.

The diagram on the next page summarizes the selection phase:

■ **Project specifications**
It sets out the broad outline of the architecture. The size of the plant, process complexity and other customer requirements decide the PLC topology, distributed I/Os architecture and connection types (see next pages).
It also covers related constraints, which can be divided into two groups:

■ **Functional constraints**
The process characteristics impose constraints in terms of power, load types and process power supply that must be complied with. This subject is introduced in the following chapter without going deeply into the calculations or protection details of LV and MV networks. Readers are advised to refer to the specific selection guides published by Schneider Electric.

■ **Operational constraints**
The project characteristics impose constraints such as:
Plant productivity: Traceabilty, environmental constraints,
Process quality: Diagnostic information required, standard to apply,
Design Cost: centralized or distributed architecture
Operation cost: operator profiles, energy monitoring,

These processes and project constraints are the inputs for the three following selection steps.

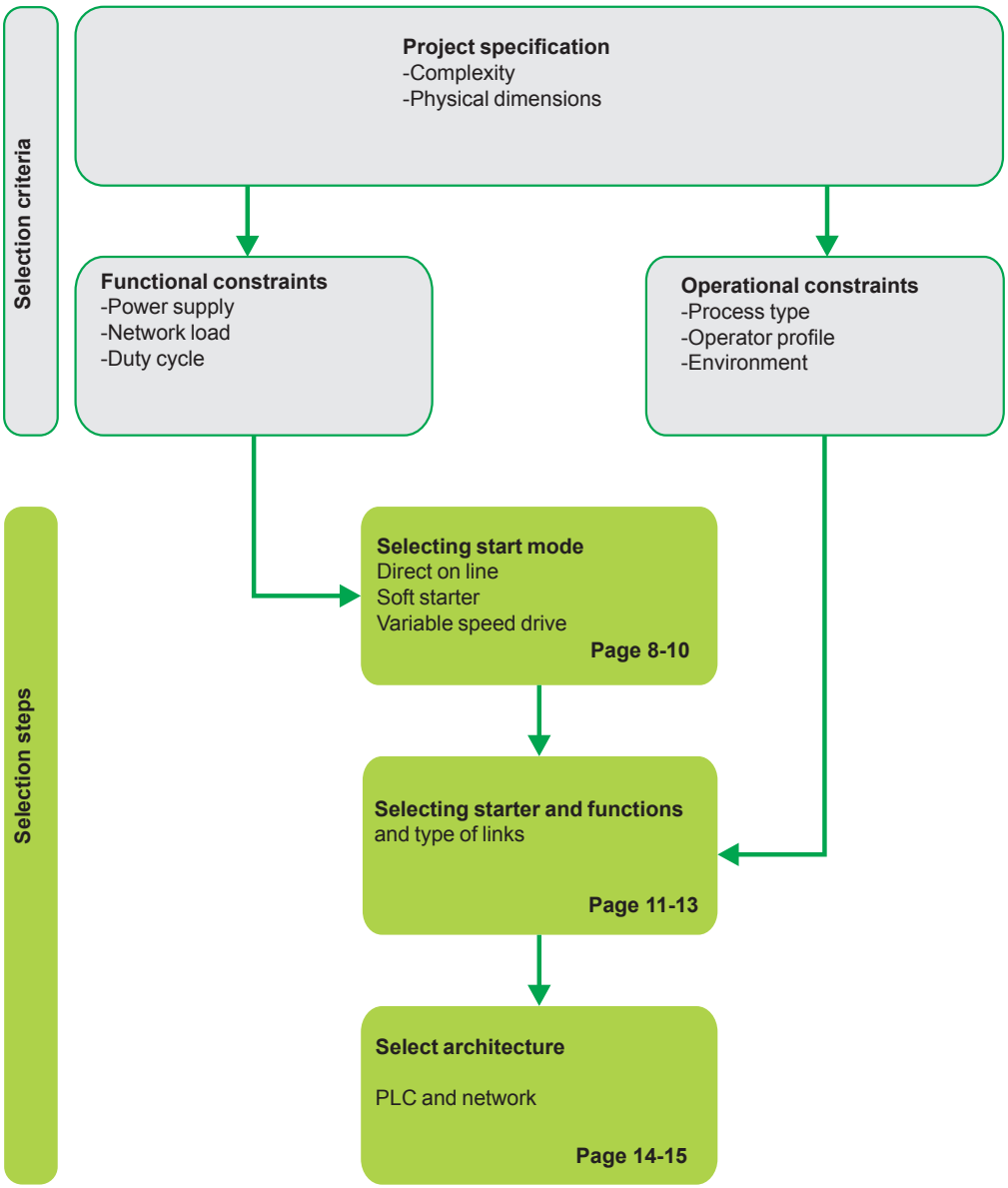
■ **Selecting starter mode:**
Three solutions are proposed to realize a motor control function:
Direct On Line starter with contactor - the simplest solution that is appropriate for low and medium power motors that do not need frequent start-up. Depending on requirements, elaborate protection and monitoring devices can be associated here.
Soft starter - it enjoys the advantage of reducing the peak starting current which can avoid water hammer and respect mechanical parts in pump applications.
Variable speed drive - the most powerful solution with an ability to regulate flows in pump applications.
It can also facilitate the optimization of energy consumption.
The starter structure is detailed on page 8 and 9. A selection guide is provided on page 10.

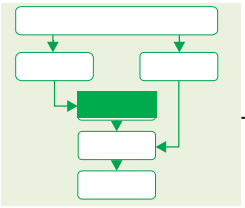
■ **Selecting motor control devices:**
The most appropriate device is selected based on requested control functions that must be provided by the starter. Those functions are categorized into three types:
Motor protection: overload and short-circuit
Metering functions: measurement of power, current, etc … .
Monitoring functions: alarms, histories, …
A table on pages 12 and 13 summarizes the various starter functions.

■ **Selecting architecture:**
All selected motor control devices have to be connected to the global system architecture. Therefore, the communication link offered by the device must be selected in a consistent manner and in compliance with the chosen global architecture.

Principle

Selection steps





Selecting starter mode

Motor starter basic functions

A motor starter unit has four basic functions:
- Isolation of the load from the main power supply,
- Protection against short-circuit,
- Protection against overload,
- Control (start, stop, speed).

Each motor starter unit can be enhanced with additional functions depending on its system requirements:
- Power: speed controller, soft starter, phase reversal, etc,
- Control: auxiliary contacts, time-delay, communication, etc.

Starter are selected based on the power and control specifications

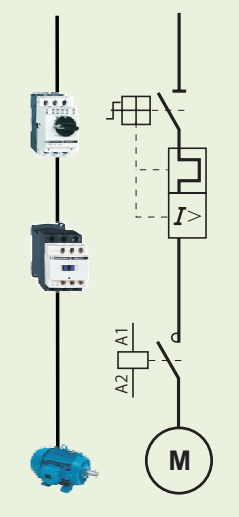
Power specifications related to the load

The choice of starter is determined by:
- Mechanical characteristics of the load (torque, inertia, speed.
- Power and electrical motor characteristics
- Necessary protections

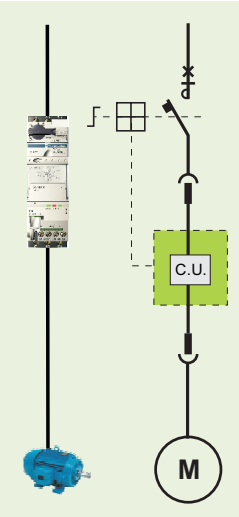
These criteria are used to define one of the following solutions
- Direct on line starter
- Progressive start-up with soft starters
- Start-up at variable speed drive (VSD)

The step for selecting and dimensioning the power part of the starter will not be developed in this document. Readers can refer to the specific guides and Schneider Electric catalogs.

Direct on line starter



■ **Direct on line starter**
This solution comprised a magneto-thermal breaker and a contactor that covers the power range up to 110kW. It provides the following basic functions:
Protection against short circuit
Protection against overload
On/Off commutator

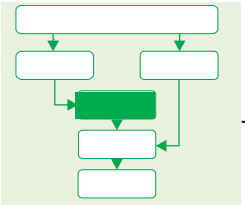


■ **TeSys U starter controller**
is an integrated Direct On Line starter up to 15kW which performs the following functions:

□ protection and control of single-phase or 3-phase motors:
- breaker function,
- overload and short-circuit protection,
- thermal overload protection and power switching,

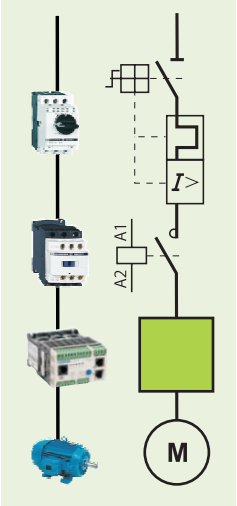
□ control of the application:
- protection-function alarms,
- application monitoring (running time, number of faults, motor current values, ...),
- logs (last 5 faults saved, together with motor parameter values).

These functions can be added by selecting control units and function modules which simply clip into the power base. The product can therefore be customised at the last moment.



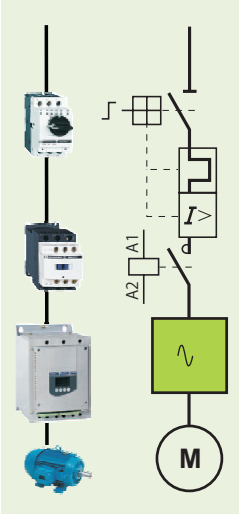
Selecting starter mode

Direct on line starter



■ **TeSys T motor management system**
The capability of over-current relay is limited when problems associated with voltage, temperature or special applications must be taken into account.
TeSys T provides complete management of the motor and its load. It incorporates below functions:
- current and voltage sensors
- hybrid analog and digital electronic technology,
- the use of communication buses for data exchange and control,
- powerful motor modelling algorithms,
- application programs whose parameters can be set.

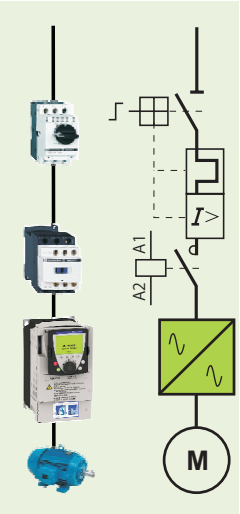
Soft starter



■ **ATS 48 Soft Starter**
Soft start - soft stop unit is a controller with 6 thyristors which is used for the torque-controlled soft starting and stopping of three-phase squirrel cage asynchronous motors in the power range between 4 and 1200 kW.
It offers soft starting and deceleration functions along with machine and motorprotection functions as well as functions for communicating with control systems.
These functions are specially designed for use in applications such as pumps, fans, conveyors, which are primarily to be found in the construction, food & beverage and chemical industries. The highperformance algorithms of the Altistart 48 contribute significantly to its robustness, safety and ease of setup.

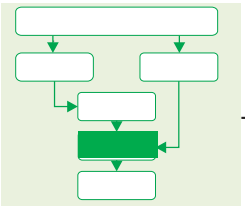
Contactor and circuit breaker could be replaced by a TeSys U controler associated with a specific module for soft starter or variable speed drive.

Variable speed drive starter



■ **ATV 31 variable speed drive**
is a frequency inverter for single and 3-phase squirrel cage asynchronous motors rated between 0.75 and 15kW.
The Altivar 31 is robust, compact and easy to use. It incorporates functions that are suitable for the most common applications, including pumps, fans, conveyors, mixers...
Modbus and CANopen protocols are integrated into the ATV31 as standard.
Altivar 31 drives are supplied with a heatsink for normal environments and ventilated enclosures. Multiple units can be mounted side by side to save space.

■ **ATV61 variable speed drive**
is a frequency inverter for 3 phase asynchronous motors rated between 0.75 and 630kW.
The Altivar 61 includes specific functions for pumping and ventilation application:
- energy saving
- automatic catching of a spinning load with speed detection
- adaptation of current limiting according to speed
- noise and resonance suppression (adjustment of switching frequency)
- Integrated PID regulator
- Electricity and service hours meterdetection of absence of fluid,
- detection of zero flow rate, limiting of flow rate
- customer settings with display of physical values (bar, l/s, °C, “etc”)
- The contactor and the circuit breaker can be replaced by a TeSys U controller.



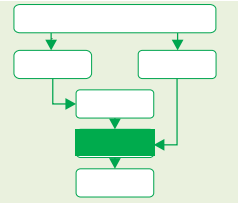
Selecting motor control devices

Introduction

The starter mode is closely linked to the load carried by the motor. The table below presents several typical applications in process control, part of which are used in processes such as water treatment or cement production. The examples illustrate how the selection is made.

Selecting devices

Type of actuator	Description/ comment	Power Range	Torque	Direct On Line	Soft starter	Speed Drive
Centrifugal Pump	. Centrifugal pumps are used to cover a wide range of volume and pressure conditions. . The flow can be controlled by using valves on the pump discharge manifold or by changing the rotation speed.	1 kW to 10 Kw 10kW to >1MW	quadratic	●	● ●	● ●
Dosing pump	. Dosing pumps are frequently used to inject fluids that may be difficult to mix efficiently in batch-tank systems because of their low volume.	< 10 kW	Constant	●		●
Screw pump	. Screw pumps are also known as Archimedes' screw. . They are used for lifting large volumes of fluid or material to a limited height. . They are driven through a speed reduction gear	1 to 50 kW	Constant	●	●	
Mixer	. Mixers are used to give homogeneity to fluids. . Agitation is also used to speed up chemical process. . Mixing is performed by a propeller rotating in the fluid driven by a speed reduction gear	1 to 50 kW	Constant		●	●
Moving devices	. Moving devices drive various types of mechanical systems such as: rotators, scrapers, shields, compressors, conveyors	1 to 10kW	Constant	●		●
Air blower and fan	. Air blowers or fans are used to provide air or oxygen for ventilation or aeration tank Flow can be adjusted using a mechanical system (fixed speed) or variable speed drive. Energy savings are possible by operating at reduced speed.	10 kW to 1 MW	Quadratic or constant		●	●
Mill and crusher	Mills and crushers are used to grind materials. They are typically high torque.	50kW to 2 MW	variable		●	●



Selecting motor control devices

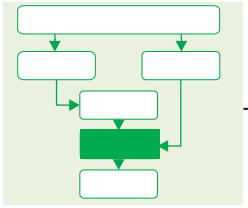
Control starter functions

Depending on needs, it is necessary to control some or all functions of a starter. The principal function groups are:

Motor control performance
Control on power, torque, speed, reversing, start time, and risk of jamming are required.. Next table summarizes the main characteristics of pumps found in process applications.

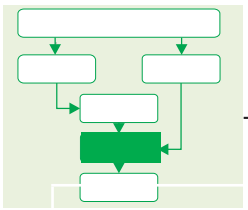
Motor protection
Its purpose is to avoid operating motors in abnormal conditions which could result in negative events such as: overheating, premature ageing, destruction of electrical windings, damage to coupling or gear box,.

Motor metering and monitoring functions
The purpose of implementing measurement devices is to ensure continuous supervision of motor operating conditions. The collected data can be used with great benefit on improving energy efficiency and extending motor lifetime.
Monitoring functions allow you to control costs, schedule maintenance operations and keep historical information for legal requirements.
The table on the next page presents a synthesis of different device functions.



Selecting motor control devices

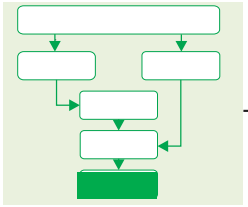
		D.O.L. starter			
		Motor circuit breaker + Contactor LC.D or F	Starter controller TeSys U		
			Standard control unit	Advanced control unit	Multifunction control unit
Motor protection functions	Short circuit	●	●	●	●
	Overload	●	●	●	●
	Locked rotor	●	●	●	●
	No load running			●	●
	Earth fault			●	●
	Supply phases failure and imbalance	●	●	●	●
	Ventilation fault				
	Abnormal temperature rise				
	Shaft bearing seizure				
	Insulation fault				
	Long starting time	●	●	●	●
	Current phase reversal				
	load fluctuations (I, U, P),				
	Overtorque				
Metering functions	Indication of motor load			●	●
	Current on 3 phases (rms value)				●
	Average current				●
	Thermal capacity level				
	Motor temperature				
	Voltages on 3 phases				
	Frequency				
	Active power, power factor				
	Earth current				
	Motor torque				
Monitoring functions	Fault differentiation			●	●
	Remote or automatic thermal reset			●	●
	Local control, with I/O on product				●
	Local control, with HMI terminal				●
	Acceleration, decelerating torque control				
	Linear, S, U or customized acceleration and deceleration ramps				
	Bypass by contactor at starting end				
	Brake sequence				
	Automatic catching a spinning load, speed detection and automatic restart				
	Energy saving ratio, 2-point or 5-point quadratic ratio				
	Preset speed				
	Adaptation of current limiting according to speed				
	Noise and resonance suppression by switching frequency				
	Electricity and service hours meter				
	Detection of absence of fluid, detection of zero flow rate, limiting flow rate				
	Sleep function, wake-up function				
	Customer settings with display of physical values: bar, l/s, °C, etc.				
	Saferty function, integrated "power removal" SIL2				
	PI regulator and reference				
	fault statistics: counters and history per type of protection,				●
	Motor statistics: storage of motor statistics values,				
	Diagnosis of faults affecting correct operation of the product.				●
	Download and save configuration				



Selecting motor control devices

		DOL	Soft starter	VSD	
		Motor management system TeSys T	ATS48	ATV31	ATV61
Motor protection functions	Short circuit		by upstream CB		
	Overload	●	●	●	●
	Locked rotor	●	●	●	●
	No load running	●	●	●	●
	Earth fault	●		●	●
	Supply phases failure and imbalance	●	●	●	●
	Ventilation fault	*	*		*
	Abnormal temperature rise	*	*		*
	Shaft bearing seizure	*	*		*
	Insulation fault	●			●
	Long starting time	●	●	●	●
	Current phase reversal	●	●	●	●
	load fluctuations (I, U, P),	●	●		●
	Overtorque		●	●	●
Metering functions	Indication of motor load	●	●	●	●
	Current on 3 phases (rms value)	●			●
	Average current	●	●		●
	Thermal capacity level	●	●	●	●
	Motor temperature	●			●
	Voltages on 3 phases	●			
	Frequency	●			
	Active power, power factor	●	●		●
	Earth current	●			●
	Motor torque		●		●
Monitoring functions	Fault differenciation	●	●	●	●
	Remote or automatic thermal reset	●	●	●	●
	Local control, with I/O on product	●	●	●	●
	Local control, with HMI terminal	●	●	●	●
	Acceleration, decelerating torque control		●	●	●
	Linear, S, U or customized acceleration and deceleration ramps			●	●
	Bypass by contactor at starting end		●		
	Brake sequence		●	●	●
	Automatic catching spinning load, speed detection and automatic restart			●	●
	Energy saving ratio, 2-point or 5-point quadratic ratio				●
	Preset speed			●	●
	Adaptation of current limiting according to speed			●	●
	Noise and resonance suppression by switching frequency			●	●
	Electricity and service hours meter		●		●
	Detection of absence of fluid, detection of zero flow rate, limiting flow rate				●
	Sleep function, wake-up function				●
	Customer settings with display of physical values: bar, l/s, °C, etc.				●
	Saferty function, integrated "power removal" SIL2				●
	PI regulator and reference			●	●
	Fault statistics: counters and history per type of protection,	●			
	Motor statistics: strorage of motor statistics values,	●		●	●
	Diagnosis of faults affecting correct operation of the product.	●		●	●
	Download and save configuration	●		●	●

*With external probes



Selecting architecture

Architecture introduction

The previous two steps allow the starter mode and motor control device type to be selected. It is now time to build the final architecture that allows all motor control devices to be connected to the PLC, HMI and SCADA system.
As described above, the choice of architecture depends on the requested level of monitoring, metering and diagnostics and also depends on how much consistency is required with the other parts of the system. Other selection criteria such as cost and performance can also influence the final choice.

Architecture principle

To illustrate this chapter and build the following chapters, an architecture example is provided.
The proposed architecture has been built using the following criteria:

■ **Mixed solution of motor control devices**
Motor control solutions are from the simplest to most advanced in order to describe the various monitoring and diagnostics capabilities in the following chapters.

■ **Communication consistency in the global system**
A solution that fulfils the requirements of the automatism (control and starting of pumps, put in parallel, monitoring loads...) and management (consumption, operating time, preventive maintenance...) is provided.

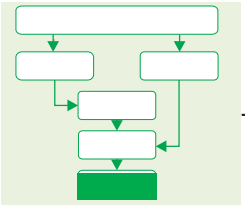
■ **Sub assembly modularity**
The designer must be able to re-use material and software for the realization of water treatment application. These subsets must be able to be duplicated for other similar applications. In order to facilitate process extension and reduce design cost, Advantys STB distributed IO islands must be used to communicate with Motor starter and variable speed drive. Advantys STB can be connected to different fieldbuses.

■ **Evolution facility**
The solution must be easily expanded beyond the initial design, so that extensions can be carried out without having to reconsider the architecture.
The table below summarises the different types of connections offered by the motor control devices.

		D.O.L. starter					Soft starter	VSD	
		Motor circuit breaker + Contactor LC.D or F	Starter controller TeSys U			Motor management system TeSys T	ATS48	ATV31	ATV61
			Standard control unit	Advanced control unit	Multifunction control unit				
Type of links	Hard wired								
	Modbus SL								
	CANopen								
	Ethernet								
	Advantys + Pre wiring								
	Advantys Internal bus								
	Advantys CANopen			*	*				

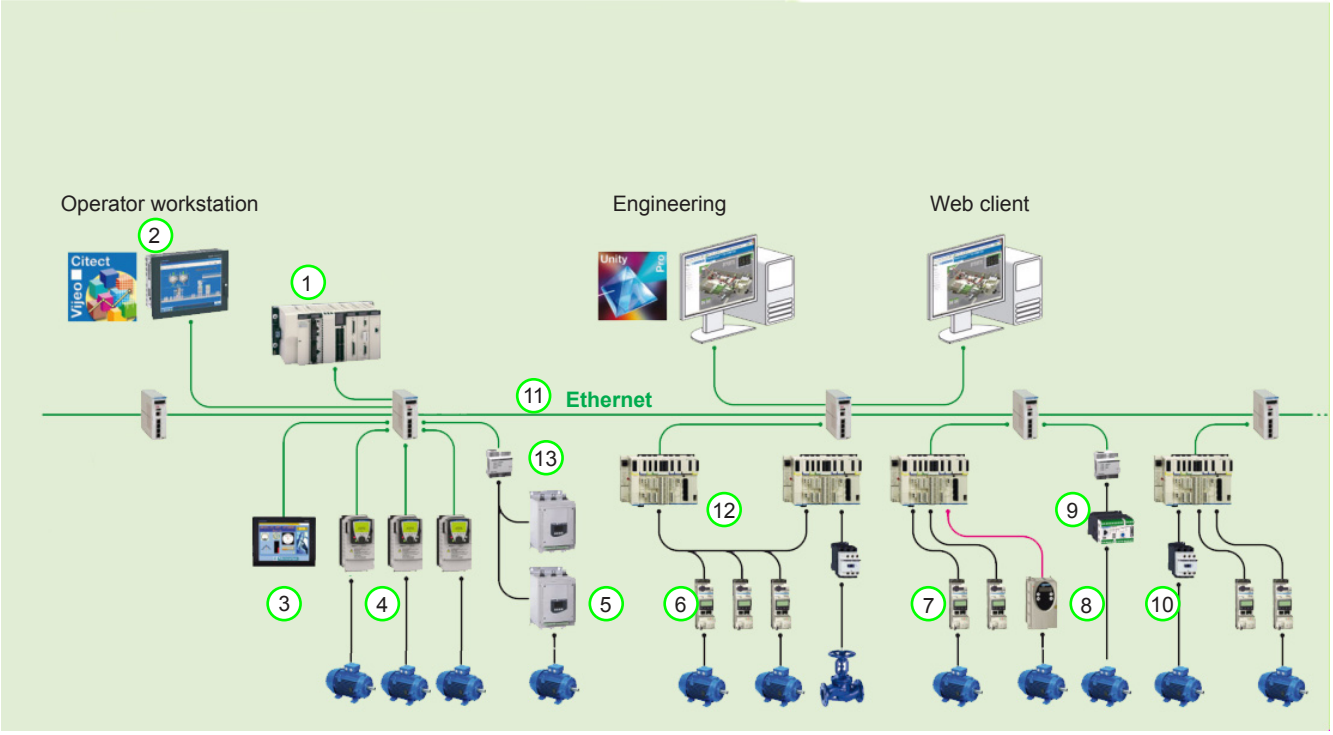
For all intelligent motor control , both Modbus and Modbus TCP with an additional Transparent Ready gateway can be proposed. CANopen is also supported by most devices.

*Available last quater of 2008



Selecting architecture

Architecture example



This architecture example is a distributed peripheral architecture with a centralized Premium PLC (1) and a standalone Vijeo Citect SCADA system (2). A Magelis HMI XBT-GT (3) is used to allow local control and monitoring.

Ethernet network (11) connects all process steps in order to allow good diagnostics and performance. Thanks to Ethernet communication, the SCADA system and HMI can easily access all process data. Embedded web diagnostic services which is available from any standard web browser facilitate the maintenance phase. The motor control devices are distributed in the plant with connection to Ethernet network.
In the same way Advantys STB islands (12) communicate with controller via Ethernet. These islands also connect DOL starter (10) and other starters through fieldbuses (6 & 8).

□ ATV 61 (4): They are directly connected to Ethernet. All monitoring functions and control are allowed. An embedded web server can be used for maintenance.

□ ATS 48 (5): A Transparent Ready gateway ETG100 (13) is used to connect this device to the architecture.

□ TeSys U with Advanced and Multifunction control unit (6): In order to reach a high level of monitoring and metering, the TeSys-U are connected to Advantys STB extension bus.

□ TeSys U with standard control unit (7): A cost effective solution is illustrated here with a pre-wired solution using the Advantys 2145 EPI module.

□ ATV31 (8): The CANopen extension of Advantys STB guarantees a cost effective connection to ATV31 with a high level of diagnostics.Remark: CAnNopen port is embedded on ATV31.

□ TeSys-T (9): A Transparent Ready gateway ETG100 is used to connect this device to the architecture. Communication transparency allows controller to get all monitoring and metering data offered by this device. The TeSys-T Ethernet port allows the gateway to be removed, which improves performance.

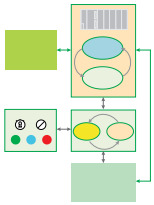
□ Contactor (10): The selected solution is wired directly to the contactors to the Advantys STB I/O module

This architecture combines various motor control device solutions to be detailed in the following chapters of this technical guide.

3-Application design

Table of contents

Operating modes	18
Hardware design	20
Software design	26



Operating modes

Introduction

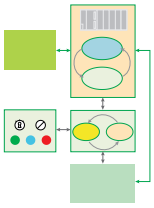
The aim of this chapter is to provide recommendations that facilitate the design phase of your process control project.
It comprises three main parts:
- Description of global application operating modes
- Description of hardware design
- Description of software design

Principle

Application operating modes are the most structuring elements in the automatism system definition phase. They determine the definition of the hardware and software part of an application and they act on all architecture components: SCADA, PLC, and motor control device wiring.
Operating modes, described below, are rather general and can be easily adapted to the specific constraints of a project.
The objective is to propose operating modes which allow starter management in remote mode with PLC or in local mode with buttons or panel.

■ **Remote mode**
In Remote Mode the motor can be controlled either by the PLC (Auto Mode) or by the SCADA/HMI operator (Manual Mode).
When both SCADA and local HMI are able to control Auto or Manual modes, the first station turning to Manual mode has the lead to manually control the actuator and to come back to Auto mode.
In Manual mode any motor can be controlled remotely from a Human Machine Interface. This operating mode can be considered risky in some targeted applications; therefore, this proposed solution can be modified to meet this constraint.

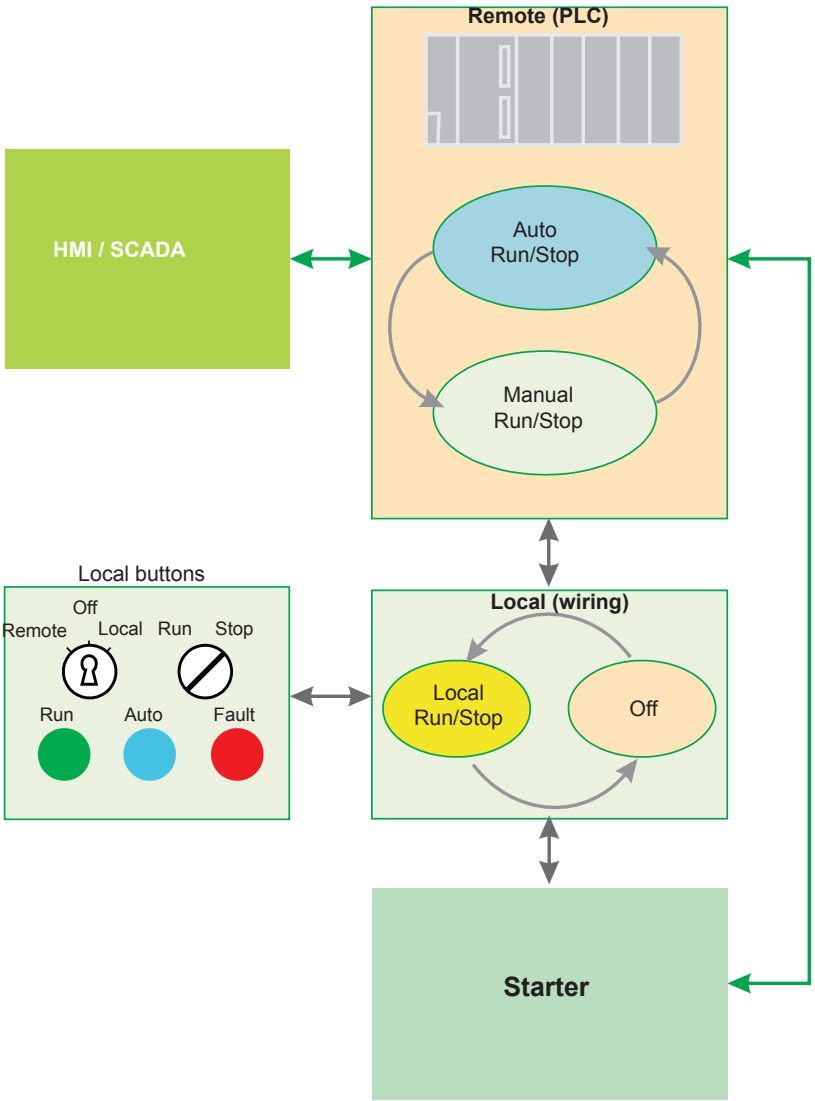
■ **Local mode**
In Local mode, an operation can be performed on actuators even in the event of a PLC fault. That means that local controls are hardwired directly to the pre-actuator (ie, to contactor or variable speed drive hardwire control circuits).
Motors can be controlled with local buttons or for variable speed drive VSD ATV61 using a Local interface.
It is possible to switch the actuator to Off mode, for maintenance purposes, in order to inform the PLC application.
For maintenance actions this mode requires an additional electrical padlocking.
When switching from Auto to Manual mode using SCADA/HMI, actuators keep its state (that means go on running if it was running previously, and keep the same speed for motors controlled by variable speed drives)
The power can be switched off by a switch. For all these modes, the emergency button is active.

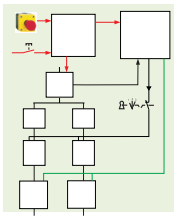


Operating modes

Principle (cont'd)

STATE		ACTORS	ACTIONS
Remote	Auto	PLC application	Run/Stop Parameters modification
	Manual	SCADA / HMI	Run/Stop Parameters modification
Local	Off	Buttons and contactor	De-energized starter
	Local (run/stop)	Buttons on starter	Run/Stop Light (fault)





Hardware design

Introduction

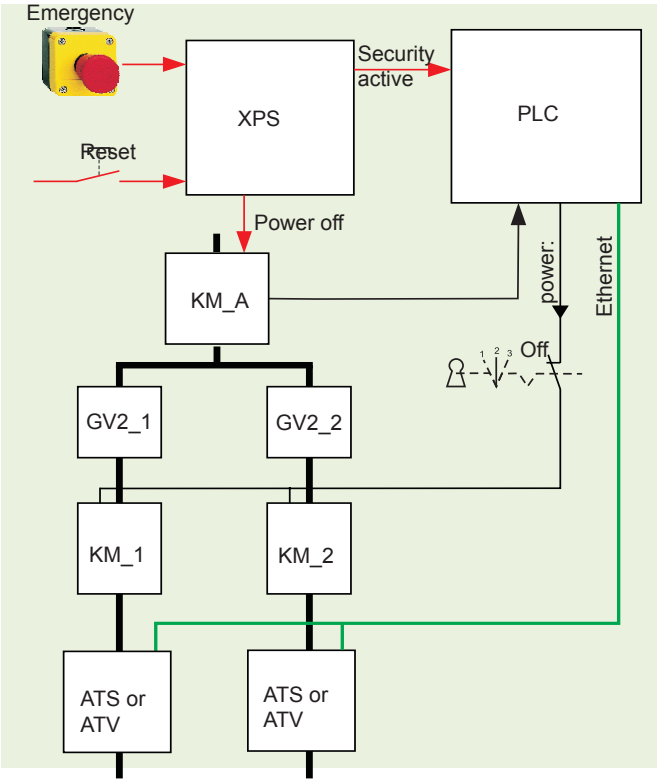
The operating mode described above requires a wiring design for an emergency stop circuit and motor control device.
The following paragraphs provide recommendations for developing a consistent wiring solution.

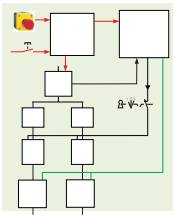
Safety requirements impose constraints to protect people and the environment. We recommend measures against electrical risks which are defined in IEC 60204-1. This standard specifies in particular the emergency stop operations.

Emergency stop circuits recommendation

The emergency stops are located in cabinets close to the machines and close to the operators. As the use of intermediate relays is prohibited, the solution of a safety function block is essential in the case of a multiple stops command.
To cover most situations, three basic diagrams are offered:

■ **Conventional diagram**
Generally it comprises a contactor, a variable speed drive, or a soft starter optionally associated with thermal protection.
A second contactor KM_A, in series with KM_1 and 2, makes it possible to cover all the categories (conforming to EN ISO 13849-1)
The PLC receives information from the safety XPS block and acts on the starter (contactors, speed variator....) via the application.
Whatever the mode, local or remote, the XPS retains priority. Resetting cannot be performed if KM1 and kM2 are closed.
This diagram covers the structures including the products:
ATS48
ATV31
TeSys-T
LRD





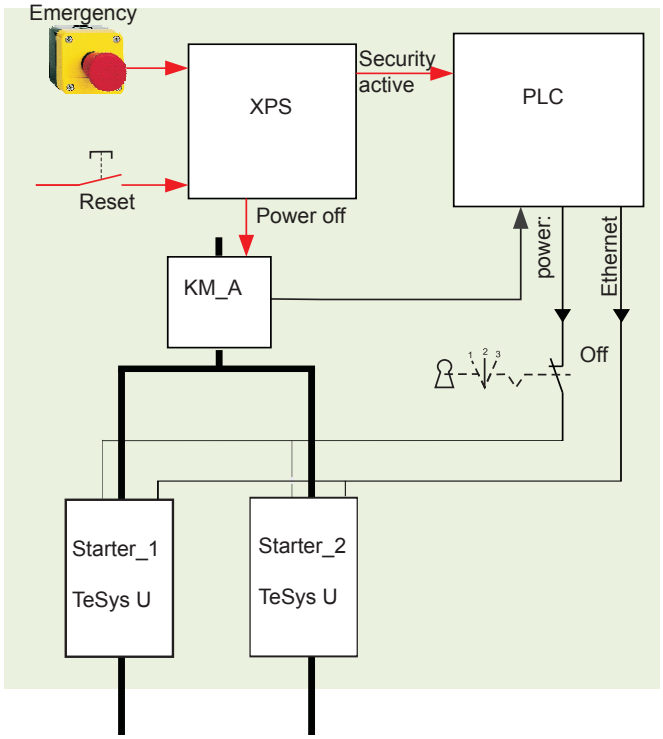
Hardware design

Emergency stop circuits recommendation (cont'd)

■ **Diagram with TeSys-U**

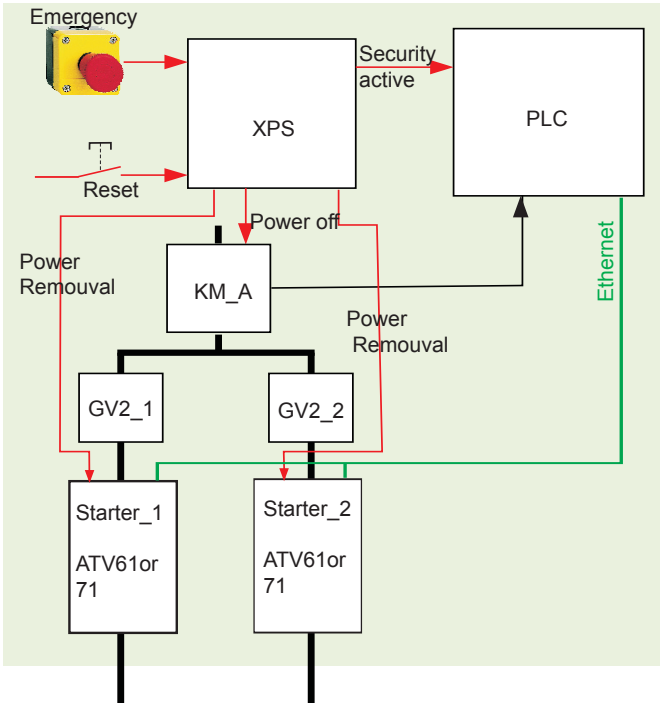
TeSys U integrates the power switching function KM1 as well as protection functions (short circuit protection, thermal protection....)

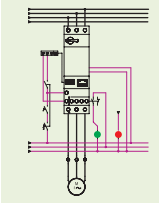
The standard requires double power breakers; contactor KM_A is mandatory.



■ **Diagram with variable speed drive ATV61 or ATV71**

In the case of a requirement not exceeding level 3 (IN ISO 13849-1) contactor KM_1 is not necessary, the power is shut down by function Power Removal (PWR) directly wirely on ATV 61/71.





Hardware design

Motor control device wiring diagrams

This paragraph provides a wiring diagram for the following starters
- **Direct on line starting TeSys U**
- **Progressive starting with soft starters**
- **Starting at variable speed with variable speed drive (VSD) ATV61**

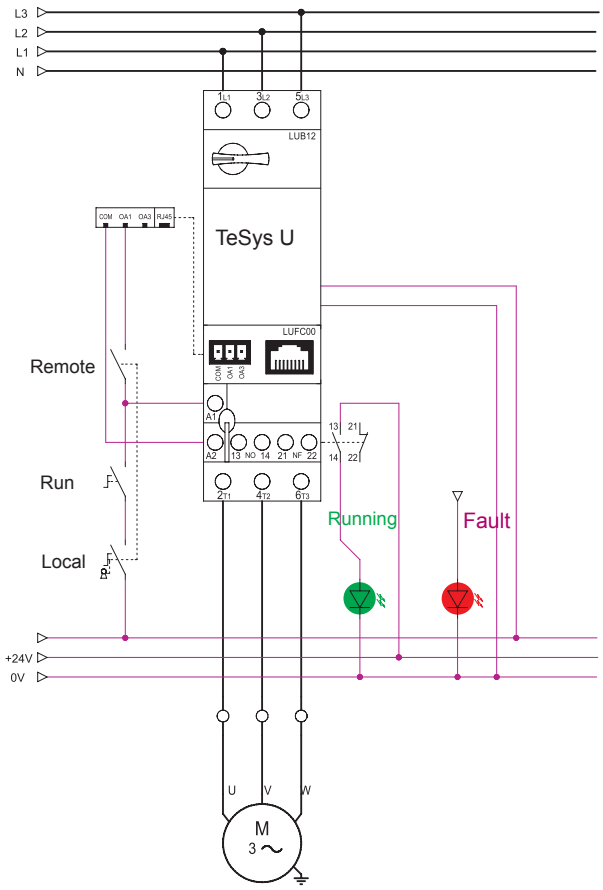
The proposed diagram re-uses previously defined operation modes. They are an extract from the water treatment application described in chapter 7. Overall diagrams for motors and pumps are available on the website of Schneider Electric.

Direct on line TeSys U diagram

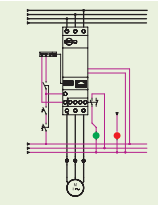
TeSys_U is a starter that integrates sectioning, protection, overload, short circuit, and commutator functions.
The commutator (selector switch) allows TeSysU to be controlled, either by the PLC in Remote mode, or by stop and start buttons in Local mode.
This lockable commutator has a 3rd position, which is Off. In this position the contactor is open and there is no longer power to the spool.
Be careful, this Off mode cannot be considered to be a padlock function.
The starter can be controlled via the Run/Stop switch; the Run command is wired directly to the starter. In this mode, the PLC is no longer in the circuit, which is important in the event of spurious signals and for maintenance.

- **Notes:**

Indicator lights wired to the starter allow Run and Fault states to be displayed.
The status of switch (local / remote) is available on PLC.
It is not possible to use the coil pre-wiring accessories to perform a manual command.
TeSys U illustrated below can be connected to Advantys STB



Hardware design



Soft starter ATS48 diagram

In the diagram below, a 116KM4 contactor is placed upstream of the ATS48. It allows the power to the starter to be cut. The switch allows the contactor to be controlled, either by PLC in Remote mode, or by start and stop buttons in Local mode.

■ **Local Mode**

The Remote/Local lockable commutator includes a 3rd position, Off, which opens the control to the contactor, which cuts the power to the terminals of the soft starter.

Be careful, this Off mode cannot be considered to be a padlocking function.

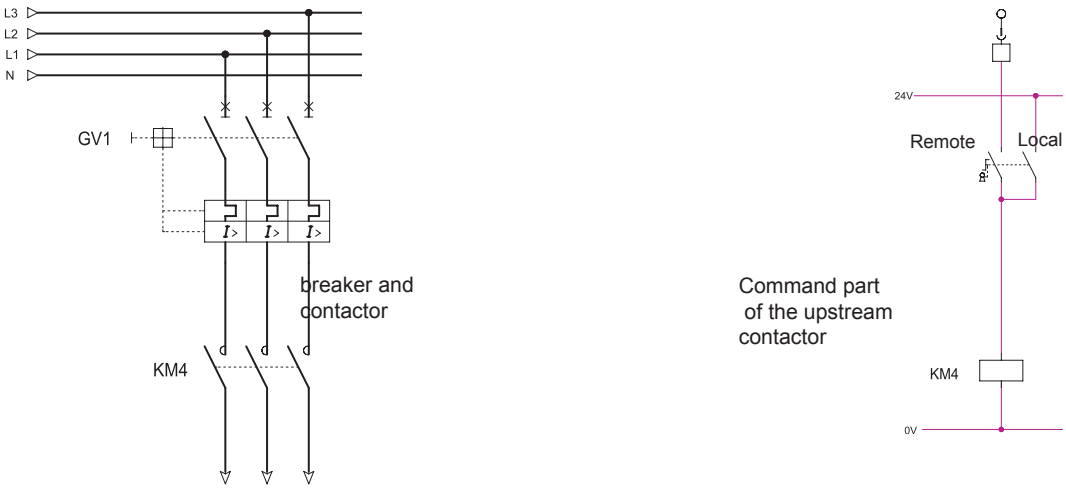
The starter can be controlled by a Run/Stop commutator; the Run command is wired directly to the soft starter. In this mode, the PLC is no longer in the circuit, which is important in the event of spurious signals and for maintenance.

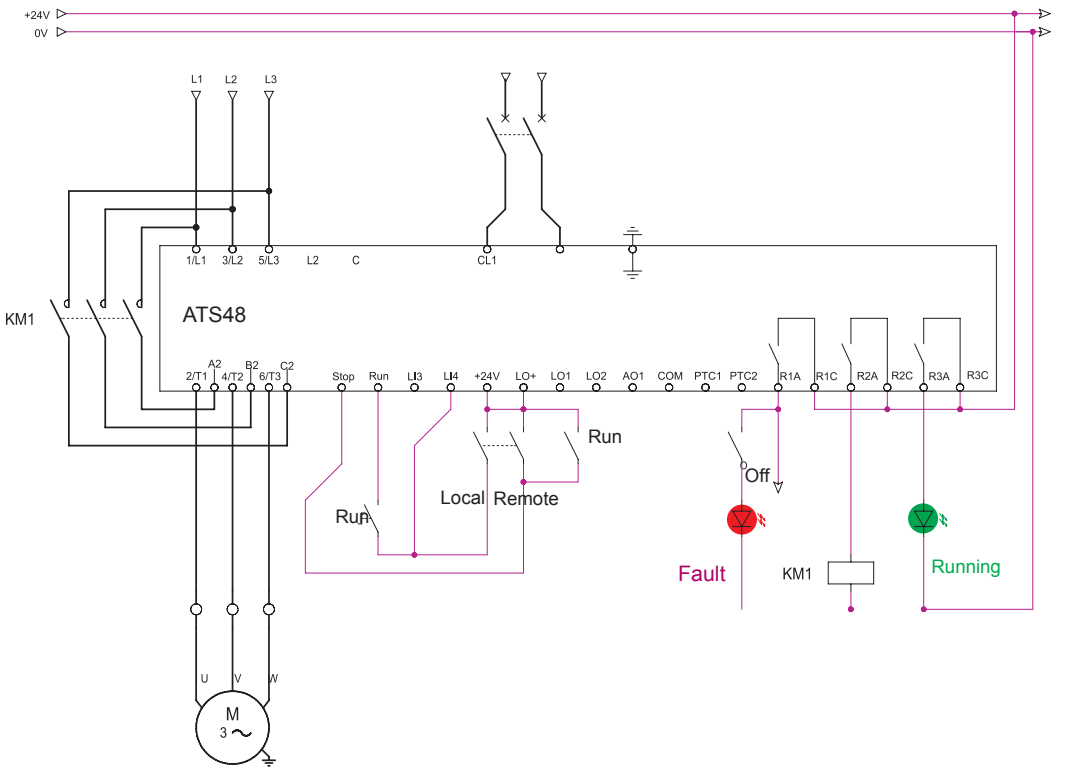
■ **Notes**

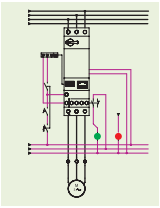
Indicator lights wired to the PLC allow Run and Fault states to be displayed.

The Remote and Local positions of the lockable commutator are repeated on the PLC.

A short circuit contactor 11KM1 is controlled by output R2 at the end of the start. To configure the I/O of the regulator the following software is recommended: PowerSuite (for PC) or ATS Display.







Hardware design

Variable speed drive ATV61 diagram

A KM1 contactor upstream of the speed regulator allows the power to the regulator to be cut (power and control diagram page 20). This contactor can be controlled, either by PLC in Remote mode, or by the commutator (selector switch) in Local mode.

■ **Local Mode**

The Remote/Local lockable commutator includes a 3rd position, Off, which opens the control to the contactor, which cuts the power to the terminals of the soft starter.

Be careful, this Off mode cannot be considered to be a padlocking function.

The starter can be controlled via the Run/Stop commutator; the Run command is wired directly to the regulator. In this mode, the PLC is no longer in the circuit, which is important in the event of spurious signals and for maintenance.

■ **Notes**

Indicator lights wired to the PLC allow Run and Fault states to be displayed.

The Remote and Local positions of the lockable commutator are repeated on the PLC.

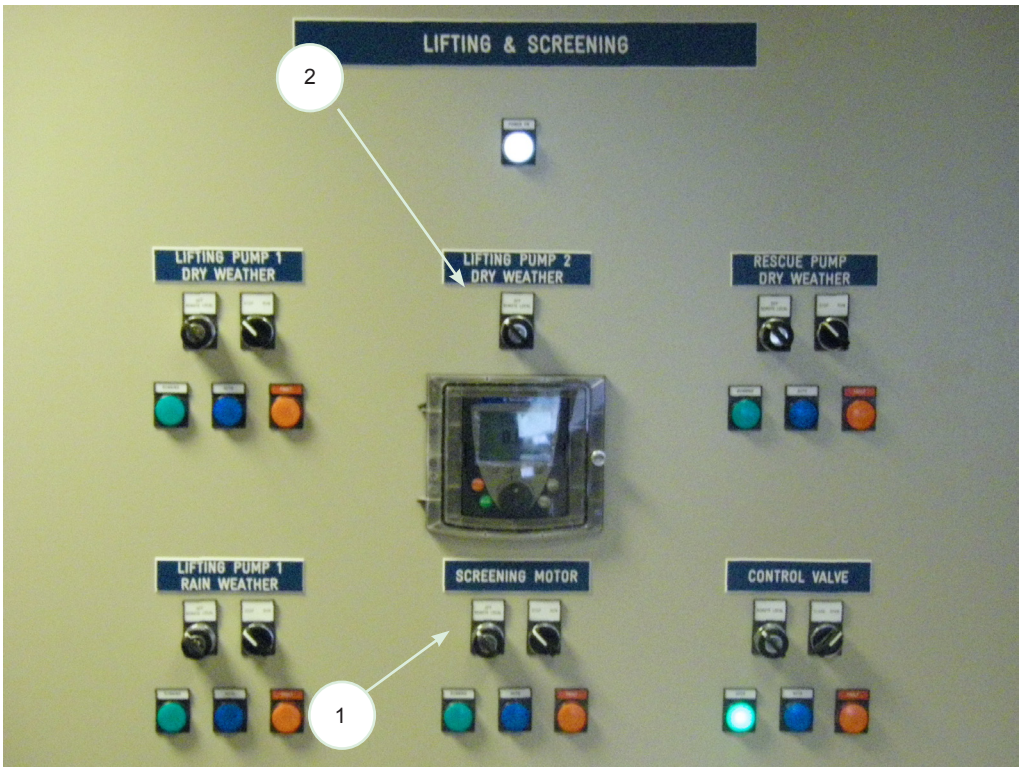
To configure the I/O of the regulator the following software is recommended: PowerSuite (for PC) or Graphic pocket (Pocket PC).

Depending on the level of security required (see page 21) a wiring alternative is possible by using the "Power Removal" input on the regulator. This type of wiring allows the KM1 contactor to be secured upstream of the regulator.

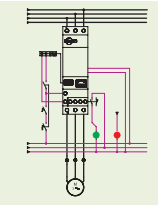
The photo above illustrates several possible variants of local command.

1 A "Remote Off Local" commutator associated with independent "Run Stop" buttons.

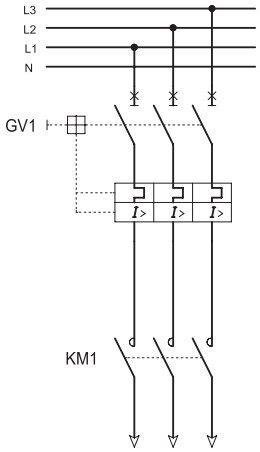
2 A "Remote Off Local" commutator associated with independent "Run Stop" buttons integrated into the regulator display.



Hardware design

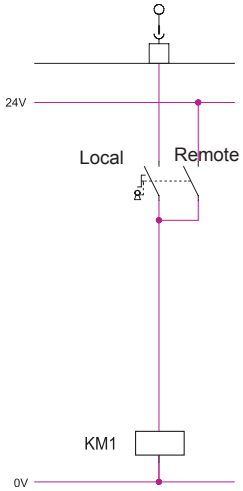


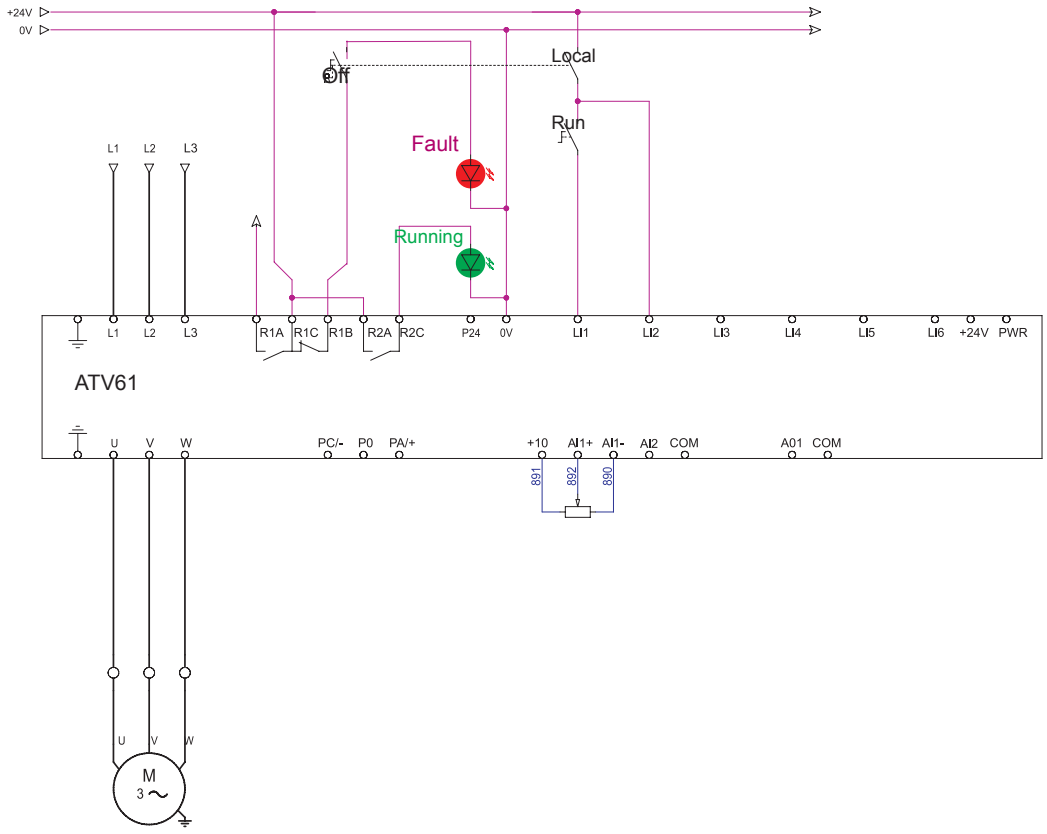
Variable speed drive ATV61 diagram (cont'd)

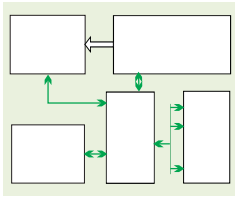


breaker and contacter upstream

Command part of the upstream contactor







Software design

Software design introduction

Motor control applications require the design of objects located in different devices using various software:
- SCADA application with Vijeo Citect V7.0
- HMI application with Vijeo Designer V4.6
- PLC application with UNITY Pro V3.1

The main components of the architecture need to exchange data and data type during build time to have a consistent application, and data during run time to execute effective and complete process control.

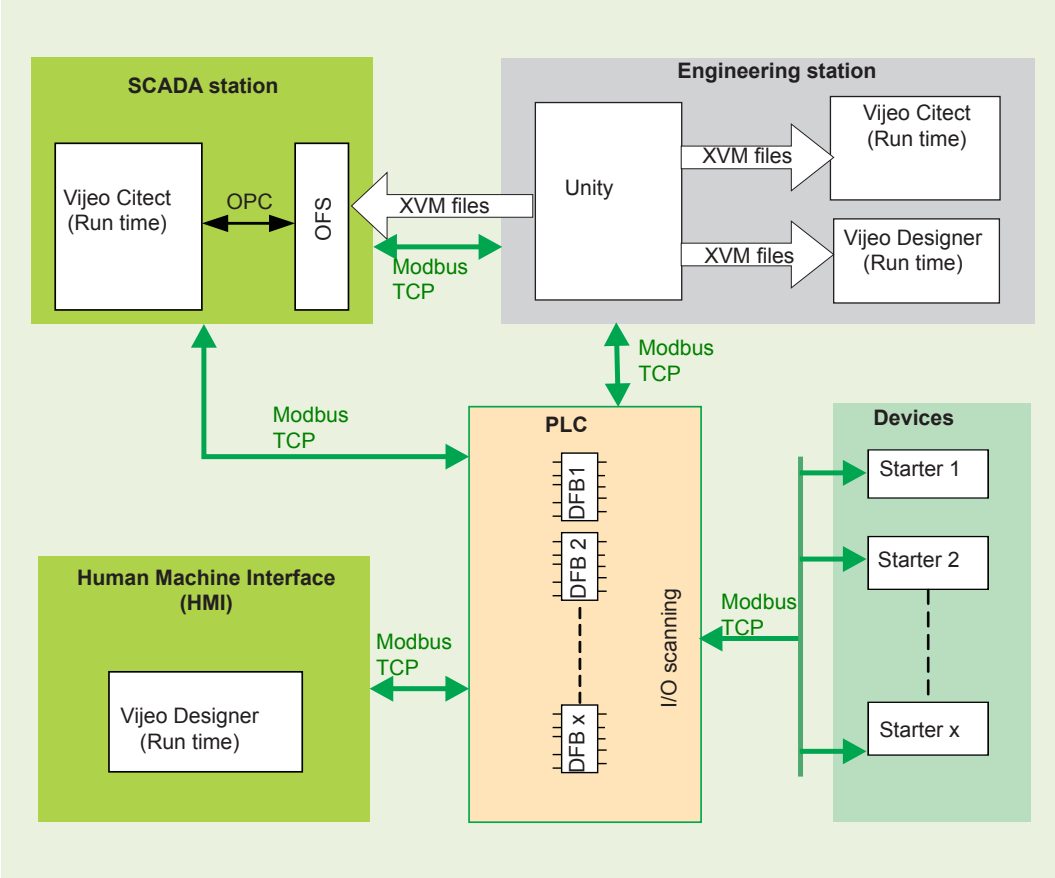
Principle

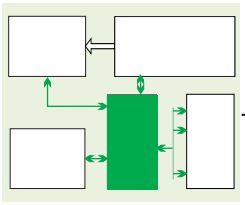
■ **On PLC application (Unity)**
The management of each pump and motor is represented by a Derived Function Blocks (DFBs)

■ **On SCADA application (Vijeo Citect)**
The human machine interface related to a pump or a motor is based on object oriented technology (Genies and Super Genies)

■ **On HMI application (Vijeo Designer)**
Basic windows and popup windows characterise motor and pump management

The choices will be detailed in the following chapters.





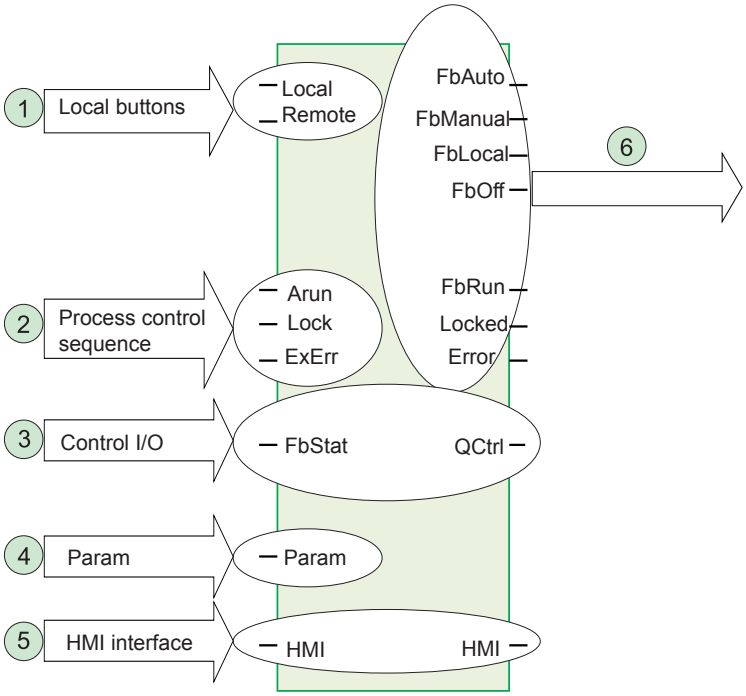
Software design

PLC and motor control device design

The exchange between PLC application and motor control devices is designed using DFB. A DFB (Derived Function Block) represents a type of starter and it is associated to the type of interface; it is used to manage:
- the operating modes: Local buttons and HMI interfaces
- the interface with the starter: Control I/O
- the human/machine interface: HMI interfaces
- the interface with the sequences and the process status: Process control sequence and Status feedback.
- the adjustment of parameters: Param.

DFB design

The general structure of the proposed DFB interface is described below. In order to have clear interfaces, the same general structure will be applied to all device DFBs.

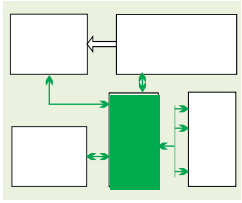


1 Local / buttons
Operating mode selection between Local/Off/Remote selector switch position input (DI).

Local	Sets the block to local mode. The pump is directly piloted by the local button box; the commands from HMI and process commands are ignored.
Remote	The PLC manages the motor. The commands come from a process sequence in Auto mode and from HMI or SCADA in Manual mode.

2 Process control sequence
This group gathers DFB inputs used by PLC program to control the device, mainly in auto mode.
Note re external errors: signals that prevent or stop device operation. Main difference to the interlock is that the external error will need to be reset via SCADA/HMI before authorizing a new start of the device. It is the user's responsibility to define the error cases that generate a fault.

ARun	Runs the motor forward in automatic mode (signal set to 1).
Lock	Interlock input for motor operation. Motor operation is stopped or inhibited, when the input is set to 1. The motor is automatically restarted, when signal returns to 0 and the run condition is still present (ARun set to 1 in automatic mode or HMI command set to 1 in manual mode).
ExErr	Input for external error signals. Motor operation is stopped or inhibited, when the input is set to 1. The error has to be acknowledged after it is gone.



Software design

DFB design (cont'd)

3 Control I/Os
This group gathers process data inputs and outputs (I/O) used to control the device. I/O scanning, based on periodic read / write variables, allows implicit exchanges of modifiable data. This functionality should be reserved for frequently used variables; access to other variables is by explicit exchanges.

I/O Scan Input (Structure)	
Fb_Stat	Starter status word. Feedback signal from I/O scanning
Fb_RPM (*)	Variable speed drive output speed. Feedback signal from I/O scanning
FB_Meas (*)	Starter measures. Feedback signal from I/O scanning The number of measurements depends on the device type and content to set up.

I/O Scan Input (Structure)	
Q_Crtl	Starter control word, sent to the starter via I/O scanning
Q_RPM (*)	Variable speed drive setpoint speed, sent to starter via I/O scanning

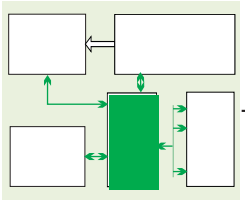
(*) Those parameters are available only on variable speed drive

4 Parameters assigment
These variables are related to the device operating mode. As they are not written as constant parameter in PLC memory, they should be saved in another way. Otherwise, on a PLC cold start all adjusted parameters would be lost.
To overcome that, we can save the parameter's current value as initial value; the following procedure could be applied:

Step	Action
A	Under Unity, validate the "save" attribute of all device parameter variables
B	Adjust device parameters at the desired value from Unity
C	With Unity, on-line mode, set system bit %S94 to 1: Initial values of all variable marked with the "save" attribute will be replaced by their current values
D	Save the application (on M340, it is necessary, in addition, to set %S66 to 1 or to transfer the RAM application to the memory card).

Following this procedure, when a PLC cold start occurs, all device parameters will be initialized with the last adjusted values.
Those DFB input parameters should be adjusted during start-up of the installation (by assigning constant value to the variables). Some of those parameters are sent to HMI or SCADA for visualization only.

Param (Structure)	
Discrepancy_time	Max time between an order and the right feedback
Min_time_Stop	Minimum time between stop and restart



Software design

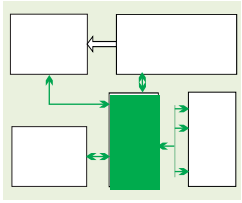
DFB design (cont'd)

5 HMI interface

This group gathers all DFB Input/Output type variables exchanged between PLC and SCADA/HMI, for the device considered.

Param (Structure)	
Auto_Man	Command to set the block in Auto or Manu
Run_Stop	Command to run or stop the pump
Clear_Fault	Command to acknowledge internal and external errors indicated at the output Error. Acknowledgement is done with a rising edge.
Clear_warning	Command to acknowledge starter warning. Acknowledgement is done with a rising edge.
Speed_Setpoint	Setpoint for motor speed in Manual mode.
Speed_Output	Output speed from the variable speed drive
Meas	Object to configure in I/O scanner and to display on HMI (eg: current, power consumption,…)
Nb_start	Number of starts performed in the last 24 hours, displayed on HMI (calculated value)
Min_time_stop	Mimimum time between stop and start
Discrepancy_time	Time minimum between a command and the right feedback
Time_to_start	Time before the motor restart, calculated when the Param_Min_time_stop<>0
Sts_Auto	Automatic mode is activated. The process sequence manages the motor
Sts_Manual	Manual mode is activated. The commands come from HMI or SCADA.
Sts_Local	Manual mode is activated. The commands are hardwired.
Sts_Off	Off mode.The power of starter is off.
Sts_Locked	Indicates that the operation is blocked by an interlock (input Lock)
Sts_Error	Indicates that the operation is blocked by an internal or external (Input Err) error, which is not acknowledged.
Sts_TimerProtect	Number start control is activated (Nb_start_day <>0) and the max starts number is reached.
Sts_Ready	Starter ready to switch on
Sts_Run	The motor is running (Status from starter)
Sts_Fwd	Motor running in positive direction
Sts_Bwd	Motor running in negative direction
Sts_Estop	Starter emergency stop activated
Sts_ExtErr	External error from the process or the system
Sts_NoVoltage_Err	No voltage on starter
Sts_Closed	Contactor closed
Sts_Tripped	Protection tripped
Sts_Butt_On	Button position "ON"
Sts_Butt_Trip	Button position "TRIP"
W_Thermal	Thermal warning
W_Module	Module warning
Sts_Ext_Err	External error from the process or the system
Sts_Discrepancy_Err	Discrepancy error. The discrepancy control is activated (Discrepancy_time<>0).
Name	Motor name & location

Note: Information in bold italic depends on starter type.



Software design

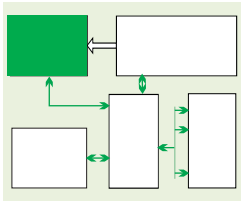
DFB design (cont'd)

The block supports automatic and manual operating modes. The automatic and manual mode are -- activated by the HMI when the mode is remote. The Manual mode is activated on PLC cold start. The local mode can also be activated by an input pin either in automatic or in manual mode. The local mode inhibits the command from process and HMI.

In the automatic mode the motor is started and stopped via the inputs ARun, if the local mode is not activated. If the operation mode is changed from automatic mode to manual mode, the motor continues in the same way, run to run, stop to stop. If the operation mode is changed from manual mode to automatic mode, the motor will follow the commands from the process.

If the operation mode is changed from Remote to Local, the motor will stop but the local hardware command will lead the stop or the run on starter terminal block. If the operation mode is changed from Local mode to Remote mode, the motor will follow the commands from the process or from HMI.
If the interlock input Lock is set to 0, the motor is running. An active interlock signal inhibits the start of the motor or stops a running motor. The motor is restarted when the interlock signal returns to 0 and the appropriate signals are still set (ARun or HMI command still on 1 or no stop by 0).

If the output Error is set to 0, the motor is running. An active interlock signal inhibits the start of the motor or stops a running motor. The function block sets the error signal, if the error input Ext_Err is set to 1 (external error) or in case of an invalid operation mode, a missing feedback signal or an internal error of the starter (internal error). The errors are indicated in the HMI as alarms. To reset the error output, the error has to be acknowledged by a rising edge on the input Ackn or by using HMI_pump structure.
If monitoring the minimum time between a stop and a start (Min_time_stop <>0), the motor will be authorized to restart only when the Time_to_start =0.



Software design

SCADA system introduction

During SCADA application build time, Vijeo Citect objects have to be defined and the data and data type associated.

- The following recommendations are provided to facilitate design, readability and re-use:
- Exchanges (figure below) are done via DDT variables (Standard Derived Data).
- Interchange file format XVM was chosen to manage the DDT variables type without needing to install the Unity tool on the station.
- OFS server (OPC Factory Server) makes it possible to use Unity structured variables in unlocated format.

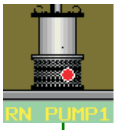
SCADA system objects

The application is based on Genie and Super Genie objects.
The Super Genies has an advantage in comparison to Popup. It is a library object, it is not linked to the page or to an application like popup. Super Genies from an application library can be reused easily by a new Vijeo Citect project using function “Link Project”.

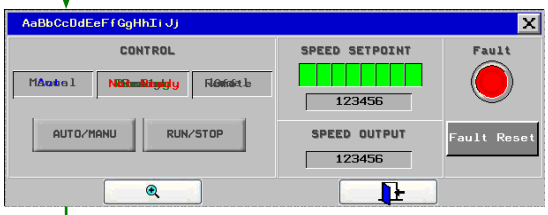
SCADA system example

Genies and Super Genies objects are associated to each type of starter element.
Genie represents a simple object such as a pump in the next figure
Super genie generates a dynamic page able to exchange variables associated with development tools.Super Genie is generally attached to Genie

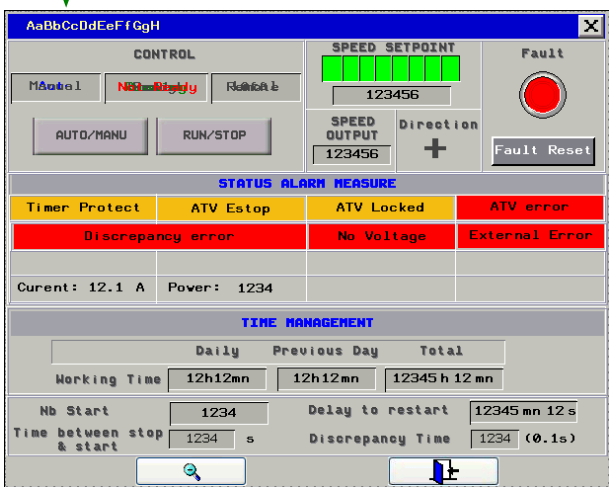
Genie

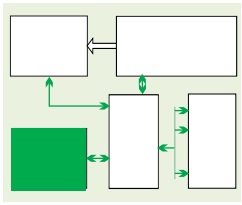


Super Genie 1



Super Genie 2





Software design

HMI system design

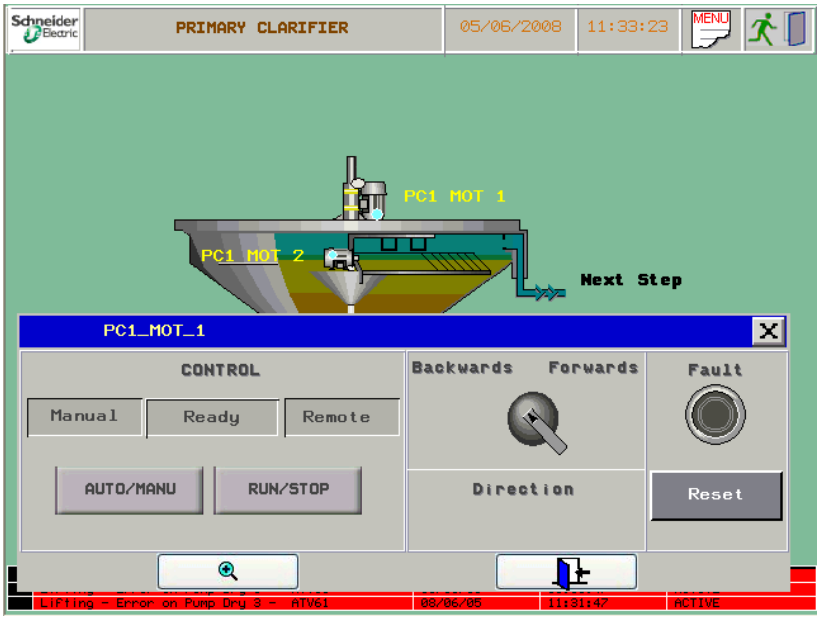
During HMI application build time, Vijeo Designer objects have to be defined and the data and data type associated.
The elementary variables (located) can be imported directly thanks to a dynamic bond in the Vijeo Designer software.The structured variables must be created manually starting from Unity description. Concrete examples are presented in the configuration chapter.

HMI system objects

HMI and SCADA interfaces are consistent with each type of starter: in the next figure, two popup Big and Small are associated
To avoid creating two popups, one Big and one Small by actuator, it is preferable to create generic popups for each actuator type.

HMI system example

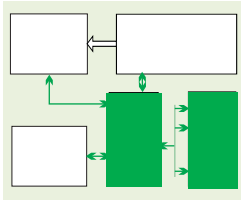
Popup objects are associated to each type of starter element.
The figure represents a simple object such as a pump



4-Configuration

Table of contents

Introduction	34
PLC and motor control device configuration	35
SCADA system configuration	40
HMI system configuration	42
Other systems configuration	42



Introduction

Purpose

The aim of this chapter is to provide key information for configuring the various system components (PLC, motor control devices, SCADA and HMI application). The main purpose is to build a consistent system configuration with the description of all data exchanged between key solution components.

The configuration of the system comprises the following stages:

- **Configuration of data to be exchanged periodically between PLC application and motor control devices**
  Different cases of I/O scanning communication service will be described in the following chapters.
- **Configuration of data and data type to be used both in PLC and SCADA applications**
  A recommendation is provided to allow a unique configuration of data in both tools.
- **Configuration of data and data type to be used both in PLC and HMI applications**
  A recommendation is provided to facilitate the configuration of data in both tools.

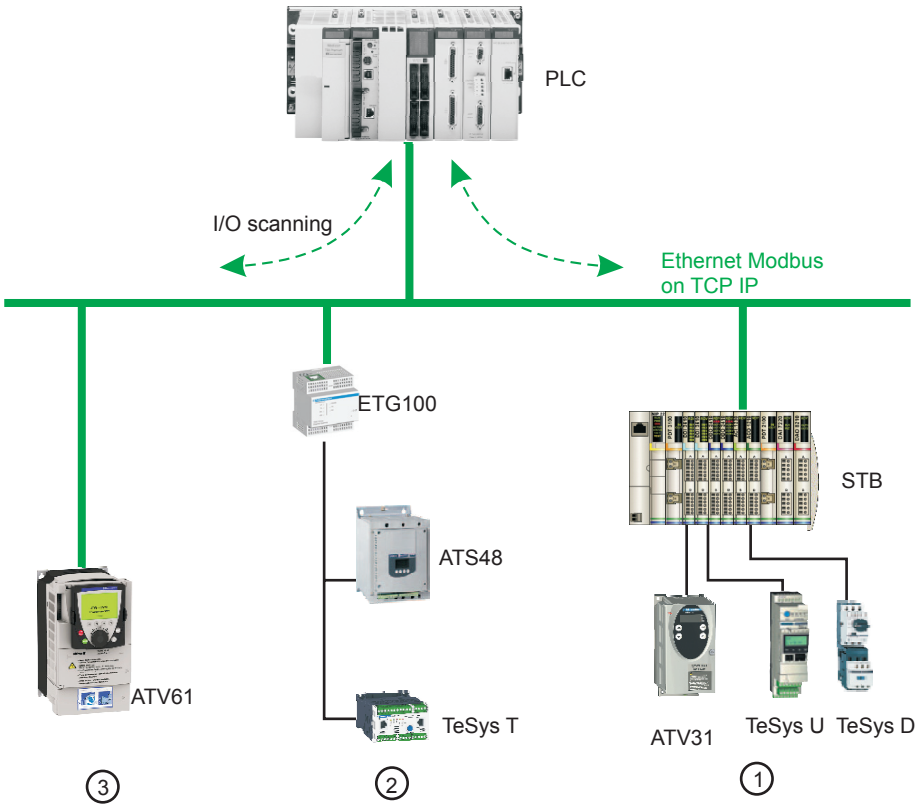
PLC and motor control device configuration

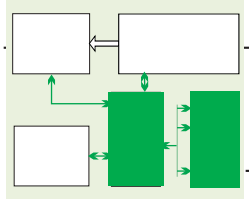
Principle

PLC application exchanges every Scan time input and output data with the motor control devices.
The I/O scanning communication service is used on Ethernet to perform this periodic communication.

To define the I/O scanning service configuration in Unity it is necessary to identify the data to be exchanged with each motor control device. In the solution platform, described in chapter 7, three types of communication architecture are used, that imply different characteristics in I/O scanning parameters.

1 Direct starter TeSys-U, TeSys D or variable speed drive ATV31 connected on Advantys STB with an Ethernet NIM (Network Interface Module) STB_NIP2212
2 Soft starter ATS_48 or TeSys T connected behind an Ethernet gateway to Modbus serial line (TSXETG100)
3 Variable speed drive ATV_61 connected directly on Ethernet





PLC and motor control device configuration

Direct starter TeSys-U configuration principle

The TeSys-U in this case is connected to Ethernet through an Advantys STB island. The connection can be done either using a pre-wiring solution or an inter-segment Advantys solution.

The configuration of motor starter data used behind an Advantys STB required two steps:

□ In the first step the data exchanged between Advantys STB and PLC must be configured in I/O scanning service. All Advantys I/O image is exchanged during this stage.

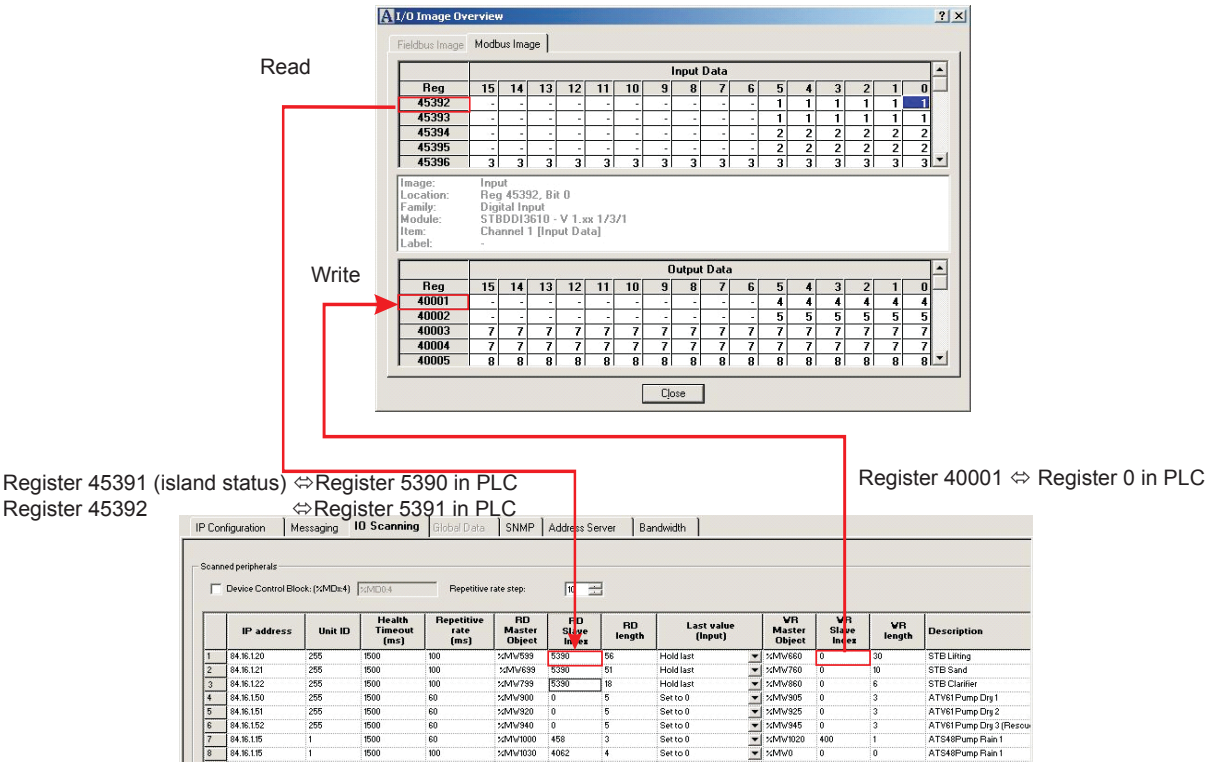
□ In the second step the data of the selected Motor starter is identified inside the Advantys I/O image. The goal is to assign a dedicated DDT to this starter.

■ **First step: I/O scanning configuration of the STB island**
Unity I/O scanning service configuration is done from the Modbus image provided by Advantys STB.
The Input base register of the Advantys Modbus image is used to fill the RD slave index of the I/O scanner (Read). In the same way, the output base register is linked to the WR slave index (Write). A first additional register is used for diagnostic purposes.

Direct starter TeSys-U configuration example

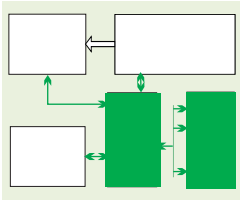
In the following example the I/O scanning service periodically reads the Modbus image of Advantys STB and stores the data from %MW599.
In the same way, data from MW%660 from Unity Pro is written to the output area of Advantys Modbus image.
Between PLC data and Advantys data a difference of one address must be taken into account during I/O scanning configuration.

Mapping objects from Advantys configurator



Mapping objects inside the PLC I/O scanning configurator

	ADVANTYS		UNITY		
	Register	Item	RD Slave index	RD Master object	Length
Input data	45391	STB island status	5390	%MW599	56
	45392	First island register input	5391	%MW600	
	45446	Last island register input	5445	%MW655	
Output data	40001	First island register input	0	%MW660	30

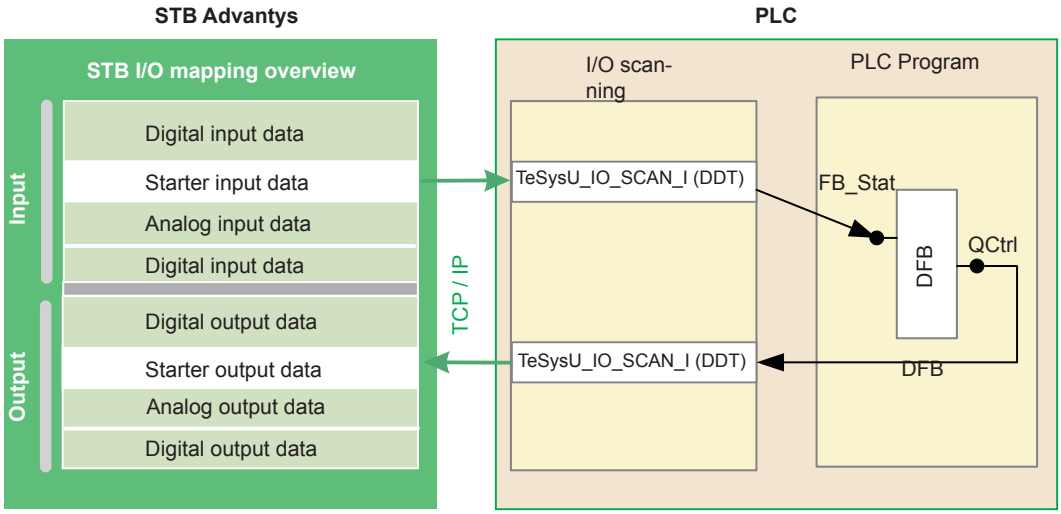


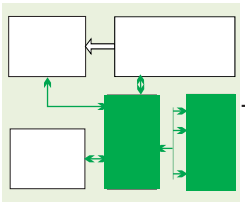
PLC and motor control device configuration

Direct starter TeSys-U configuration example

■ **Second step: Select the Motor starter data**

The input and output data of the selected motor starter must be identified within the I/O scanning data.
The Advantys STB configuration tool is used to determine the right offset that fits the starter.
A DDT is associated to the motor starter data type.
This data is used as input and output parameters (Control I/O) of the DFB linked to the motor starters.
This diagram shows the link between a device (starter) connected on STB and the PLC.
Using Advantys configurator we obtain the complete STB I/O mapping. Each I/O module is associated to registers.
After register identifications, relevant registers are mapped to the predefined structures (DDTs).. Inside the PLC application the structure is connected to the DFB pin.

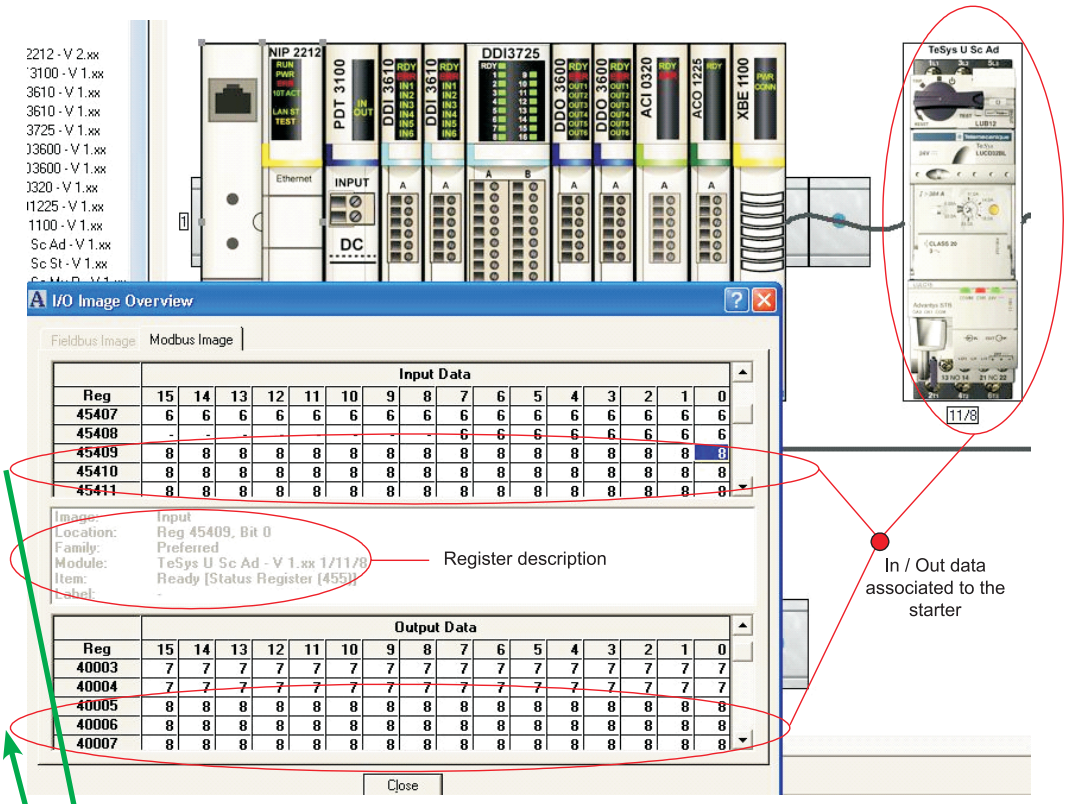




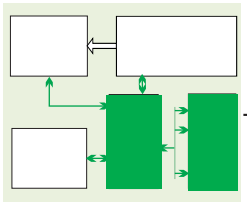
PLC and motor control device configuration

Direct starter TeSys-U configuration example

(cont'd)



	ADVANTYS		UNITY	
	Register	Item	Located address	Variable (DDT)
Input data	45409	Status 455	%MW617	Type TeSysUa_IO_SCAN_I
	45410	Status 458	%MW618	
	45411	Status 461	%MW619	
Output data	40005	Control Register	%MW664	Type TeSysUa_IO_SCAN_0
	40006	Communication Register	%MW665	
	40007	Contol Output Register	%MW666	



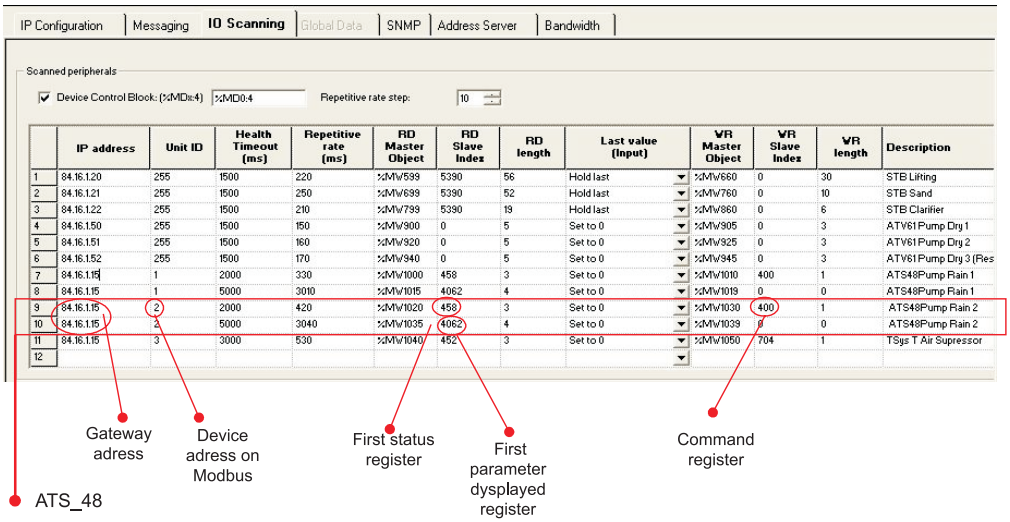
PLC and motor control device configuration

ATS48 configuration principle

The ATS48, as well as the TeSys-T is connected to an Ethernet network through an ETG100 gateway. Communication is transparent between the Ethernet and Modbus serial line. Therefore, the I/O scanning service can directly access the ATS48 data. The unit ID identifies the slave address of the soft starter on Modbus.

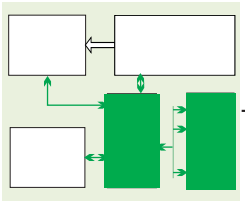
ATS48 configuration example

In the following example, three status registers and four displayed registers are read, and one command register is written (see below).
Note: Two I/O scanning lines are required to configure the input data, as status registers and displayed parameters are not in a contiguous area.


IP Configuration
Messaging
IO Scanning
Global Data
SNMP
Address Server
Bandwidth
Scanned peripherals
Device Control Block: (%MDb:4)
%MD0:4
Repetitive rate step:
10
IP address
Unit ID
Health Timeout (ms)
Repetitive rate (ms)
RD Master Object
RD Slave Index
RD length
Last value (Input)
WR Master Object
WR Slave Index
WR length
Description
1 84.16.1.20 255 1500 220 %MW599 5390 56 Hold last %MW660 0 30 STB Lifting
2 84.16.1.21 255 1500 250 %MW699 5390 52 Hold last %MW760 0 10 STB Sand
3 84.16.1.22 255 1500 210 %MW799 5390 19 Hold last %MW860 0 6 STB Clarifier
4 84.16.1.50 255 1500 150 %MW900 0 5 Set to 0 %MW905 0 3 ATV61 Pump Dry 1
5 84.16.1.51 255 1500 160 %MW920 0 5 Set to 0 %MW925 0 3 ATV61 Pump Dry 2
6 84.16.1.52 255 1500 170 %MW940 0 5 Set to 0 %MW945 0 3 ATV61 Pump Dry 3 (Res
7 84.16.1.15 1 2000 330 %MW1000 458 3 Set to 0 %MW1010 400 1 ATS48 Pump Rain 1
8 84.16.1.15 1 5000 3010 %MW1015 4062 4 Set to 0 %MW1019 0 0 ATS48 Pump Rain 1
9 84.16.1.15 2 2000 420 %MW1020 458 3 Set to 0 %MW1030 400 1 ATS48 Pump Rain 2
10 84.16.1.15 2 5000 3040 %MW1035 4062 4 Set to 0 %MW1039 0 0 ATS48 Pump Rain 2
11 84.16.1.15 3 3000 530 %MW1040 452 3 Set to 0 %MW1050 704 1 TSys T Air Supressor
12
Gateway adress
Device adress on Modbus
First status register
First parameter dysplayed register
Command register
ATS_48

	ATS		UNITY	
	Register	Item(*)	Located address	Variable (DDT)
Input data	458	ETA	%MW1000	Type ATS48_IO_SCAN_I
	459	ETI	%MW1001	
	460	ETI2	%MW1002	
	4062	LCR	%MW1035	Type ATS48_IO_SCAN_IM
	4063	LTR	%MW1036	
	4064	THR	%MW1037	
	4065	PHE	%MW1038	
Output data	400	CMD	%MW1030	Type ATS48_IO_SCAN_O

(*)Item for more details, see Implementation chapter



PLC and motor control device configuration

ATV61 configuration principle

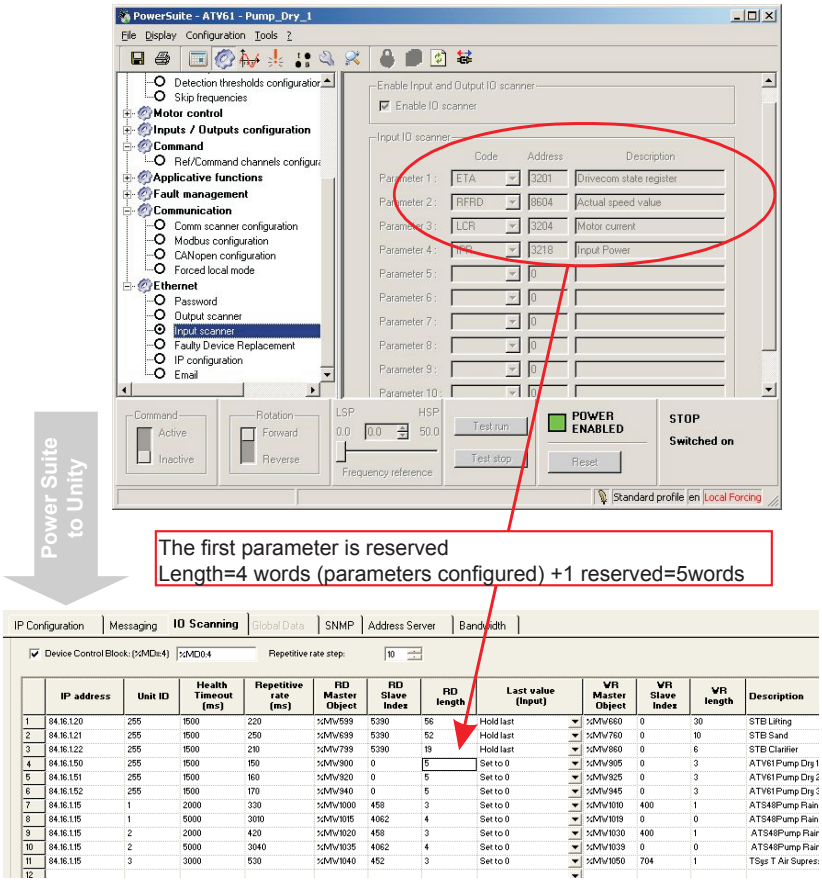
Considering the ATV61, the input and output I/O scanning parameters can be configured in different ways:

- using Power Suite software
- using the ATV_61 graphic display terminal
- using the ATV_61 Web server and Internet Explorer

In Power Suite a dedicated screen is used to declare the data exchange with PLC. A first parameter is added for status purpose. These parameters are numbered from 0.
In UNITY PLC configuration the number of parameters declared in the drive will determine the length of registers to Read and to Write with the I/O scanner.

ATV61 configuration example

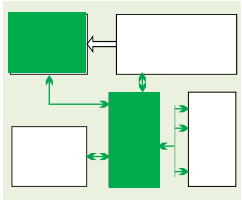
The following example presents a configuration of four input variables in Power Suite (Variable speed drive configuration) and in Unity I/O scanning service (PLC software)
Similar work is necessary for output variables.



5 data is periodically read from ATV61 and stored in PLC from %MW900.

	ATV61			UNITY	
	Register	Item	Description	Located address	Variable (DDT)
Input data		Reserved		%MW900	ATV61_IO_SCAN
	3201	ETA	State register	%MW901	
	8604	RFRD	Actual speed value	%MW902	
	3204	LCR	Motor curent	%MW903	
	3218	IPR	Input power	%MW904	
Output data	8501	CMD	Command register	%MW905	ATV61_IO_SCAN
	8602	LFRD	Target velocity	%MW906	

Notes: The speed registers RFRD and LFRD can be replaced by RFR and FRH to work with frequency unit.

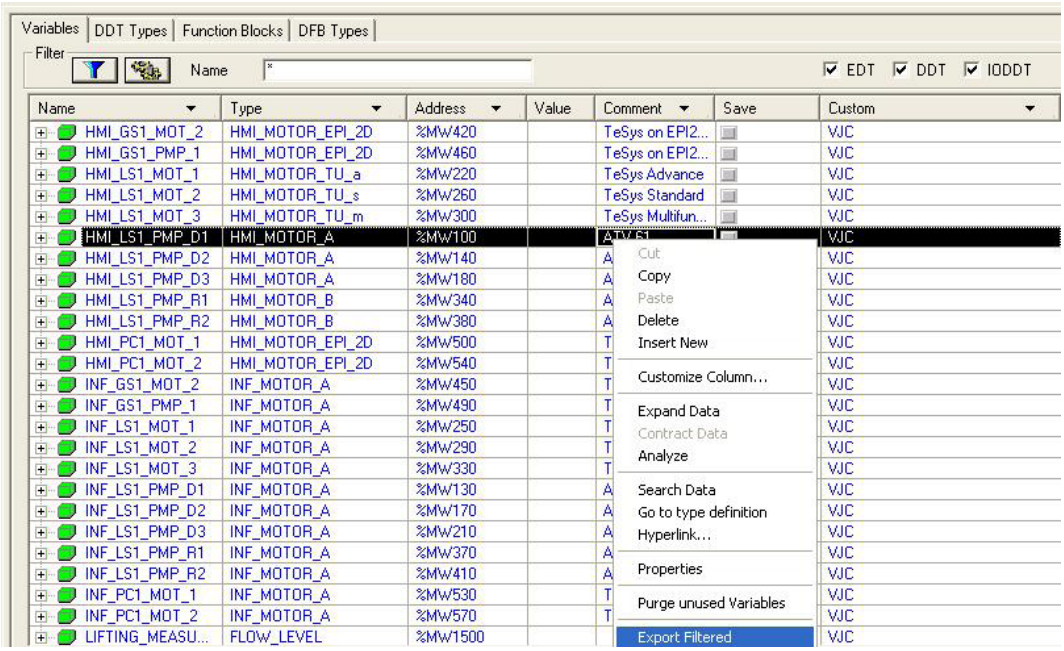


SCADA system configuration

SCADA system principle

The SCADA I/O tag database is created in Vijeo Citect from the UNITY variable database.
We recommend creating at an early stage the most complete I/O tag database in Vijeo Citect, by importing the relevant PLC tags. The goal is to have a unique tag configuration from PLC to SCADA.

■ **Database creation**
The SCADA database is created from the UNITY PLC variable database. It is advisable to create a filter in UNITY to select only the required SCADA variables.
The filtered UNITY file is then exported to an XVM file used by Vijeo Citect.
Procedure:
In the Unity application, to create the filter (see below), use the “custom” field of variables that should be exchanged with SCADA :
□ - VJA for alarms
□ - VJT for trends
□ - VJC for other variables
Remark:
To avoid variables list overload, into Citect Explorer\ Tool \ Import Tags, check off “ Purge deleted tag not found in data source.

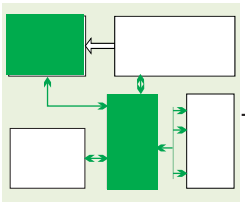


OFS server and XVM file are used for the variables exchange, the procedure in OFS is::
- declare the device
- the address of the PLC
- the name and filepath of the associated XVM file

■ **Database consistency**
During application debug, discrepancies between PLC and SCADA databases can appear. We recommend using one of the following two methods:

□ Case 1, OFS has dynamic access to the XVM file. In this case we have to check in UNITY application the option “Auto saving on download = XVM file”.
After each PLC application modification, the XVM file is updated and downloaded therefore database consistency between PLC and SCADA is guaranteed.

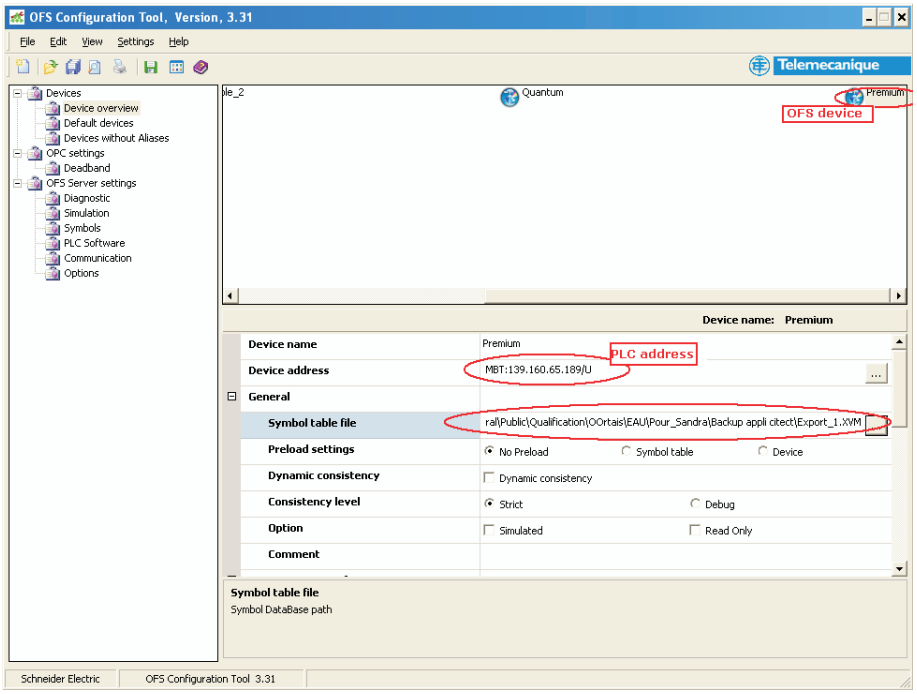
□ Case 2, OFS has no dynamic access to the XVM file. In the “Device Overview” folder of OFS configuration tool, “Consistency level” parameter must be set to “Debug” mode. In this case, OFS accepts discrepancies between PLC and SCADA.



SCADA system configuration

SCADA system principle

(cont'd)



Without describing in detail the Vijeo Citect programming, we will quote only the principle stages (see below):

- Step N°1: Creating clusters

Citect Project Editor> Servers> Clusters.

Create Cluster 1

- Step N°2: Creating the network address

Citect Project Editor> Servers Network Addresses> OFS server address

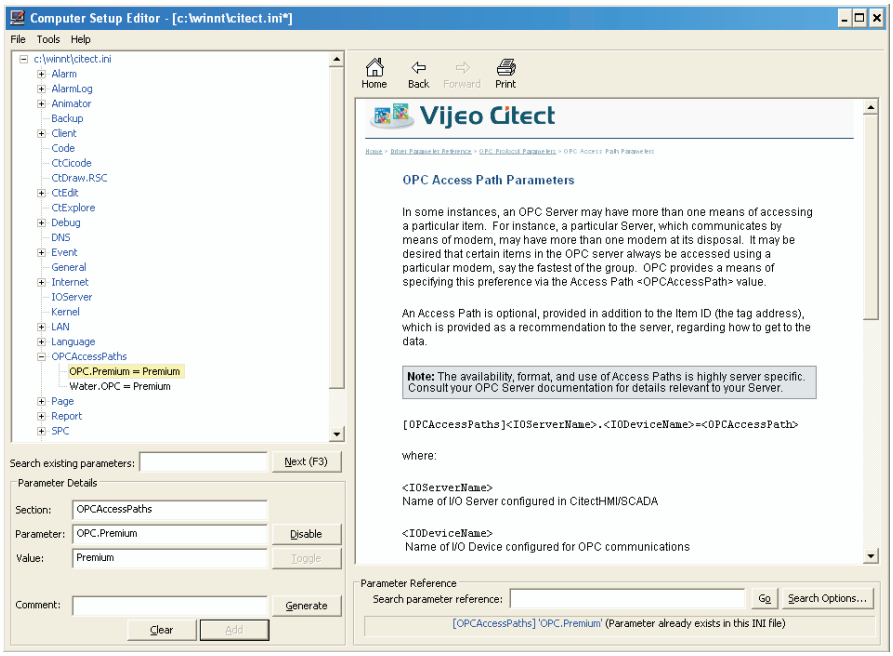
- Step N°3: Creating Servers

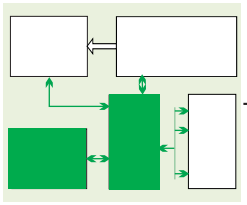
Creating Alarm and Trend servers and link to Cluster 1.

Creating the I/O server (OPC) and My I/O device (Premium): Citect Project Editor> Communication > Communication wizard use

- Step N°4: OFS Communication definition

In [OPCAccessPath] of citect.ini, added: IOServer. IODevice=OFS Device.


Computer Setup Editor - [c:\winnt\citect.ini*]
File Tools Help
c:\winnt\citect.ini
Alarm
AlarmLog
Animator
Backup
Client
Code
CtCicode
CtDraw.RSC
CtEdit
CtExplore
Debug
DNS
Event
General
Internet
IOServer
Kernel
LAN
Language
OPCAccessPaths
OPC.Premium = Premium
Water.OPC = Premium
Page
Report
SPC
Search existing parameters:
Next (F3)
Parameter Details
Section: OPCAccessPaths
Parameter: OPC.Premium
Value: Premium
Disable
Toggle
Comment:
Generate
Clear
Add
Home Back Forward Print
Vijeo Citect
OPC Access Path Parameters
In some instances, an OPC Server may have more than one means of accessing a particular item. For instance, a particular Server, which communicates by means of modem, may have more than one modem at its disposal. It may be desired that certain items in the OPC server always be accessed using a particular modem, say the fastest of the group. OPC provides a means of specifying this preference via the Access Path <OPCAccessPath> value.
An Access Path is optional, provided in addition to the Item ID (the tag address), which is provided as a recommendation to the server, regarding how to get to the data.
Note: The availability, format, and use of Access Paths is highly server specific. Consult your OPC Server documentation for details relevant to your Server.
[OPCAccessPaths]<IOServerName>.<IODeviceName>=<OPCAccessPath>
where:
<IOServerName>
Name of I/O Server configured in CitectHMI/SCADA
<IODeviceName>
Name of I/O Device configured for OPC communications
Parameter Reference
Search parameter reference:
Go
Search Options...
[OPCAccessPaths] 'OPC.Premium' (Parameter already exists in this INI file)



HMI system configuration

HMI system configuration principle

During the HMI build time the tag database has to be created from the UNITY PLC variable database.

A link between Unity XVM file and Vijeo Designer project has to be done. As Vijeo Designer variables do not access Unity structured and unlocated variables, it is necessary to recreate structured datatypes. These structure data types allow us to use generic popups using indexed physical addresses.

Other systems configuration

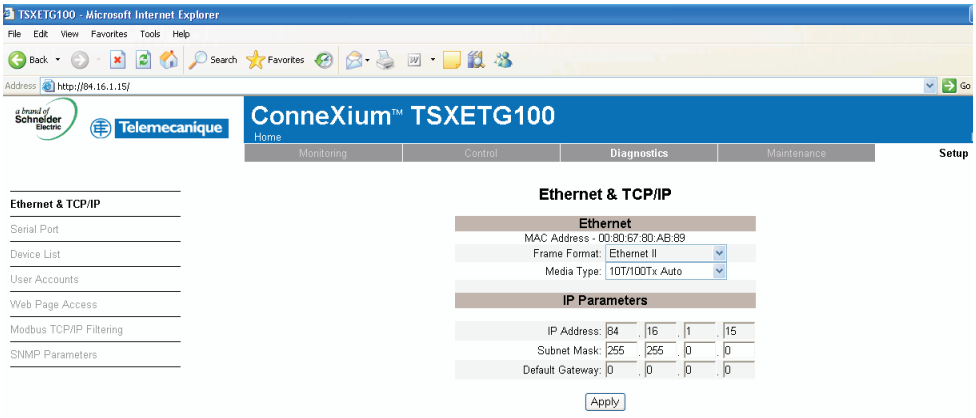
ETG100 configuration

■ **Connetion to the gateway**

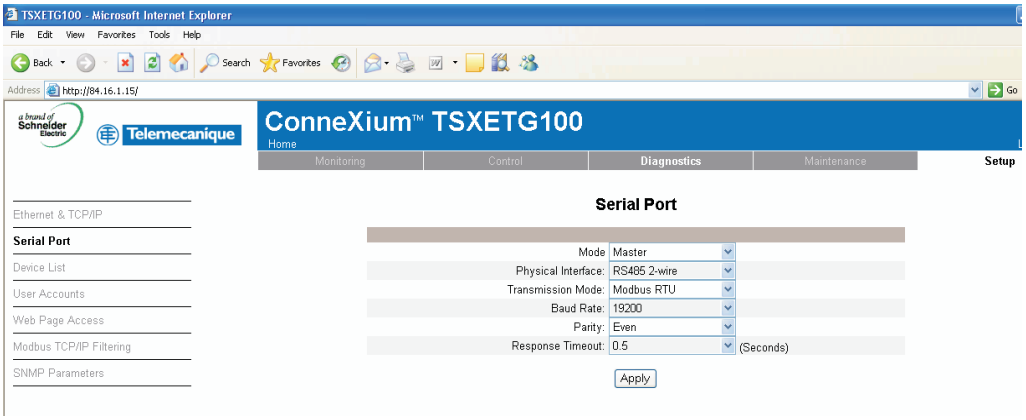
The ATS48 and TeSys-T are connected to the Ethernet network via an ETG100 gateway. Communication is transparent between the Ethernet and Modbus serial line therefore the I/O scanning service can directly access the device data. To configure this gateway, an Internet browser is connected with.
Default address (169.254.0.10)
Login “Administrator”
Password “Gateway”

■ **Configuration**

Ethernet parameters configuration



Modbus parameter configuration

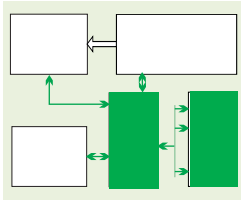


Note:
The I/O scanner sends several questions in parallel while the gateway sequences them one by one. Therefore, if a Modbus unit does not respond to a fault for example, this will generate a time-out at the gateway which will take precedence over the time-out of the I/O scanner. The result is that a Modbus unit with failed communication can trigger the time-out of the I/O scanner for the other Modbus equipment. To minimize this, it is necessary to set a minimum time-out (0.5s) on the serial port of the gateway.

5-Implementation

Table of contents

Introduction	44
PLC and motor control device implementation	45
Unity program implementation	52
SCADA program implementation	53
HMI program implementation	54



Introduction

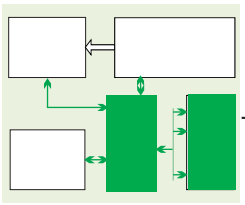
Purpose

The main purpose of this chapter is to detail how to implement the various components introduced in the Design chapter.
An example of implementation will also be described in Chapter 7 with additional information concerning programming rules and examples.

Overview

For each type of associated starter at each connection type, a dedicated DFB and data structure are implemented
The next table lists the set of objects

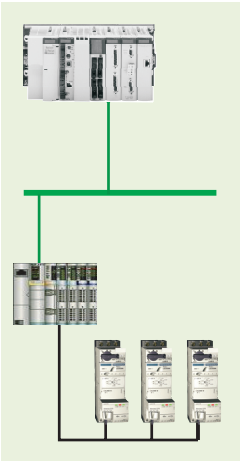
Starter	DFB	I/O scanning	Param (Structure)	Param (Structure)
ATV 61	MOT_ATV61	ATV61_IO_SCAN	HMI_MOTOR_A	PUMP_PARAM
ATV31	MOT_ATV31	ATV31_IO_SCAN_I ATV31_IO_SCAN_O	HMI_MOTOR_C	PUMP_PARAM
ATS48	MOT_ATS48	ATS48_IO_SCAN_I ATS48_IO_SCAN_IM- ATS48_IO_SCAN_O	HMI_MOTOR_B	PUMP_PARAM
TeSys U std (com)	MOT_TeSysU_s	TeSysUs_IO_SCAN_I TeSysU_IO_SCAN_O	HMI_MOTOR_TU_s	PUMP_PARAM
TeSys U std (com)	MOT_TeSysU_a	TeSysUa_IO_SCAN_I TeSysU_IO_SCAN_O	HMI_MOTOR_TU_a	PUMP_PARAM
TeSys U std (com)	MOT_TeSysU_m	TeSysUm_IO_SCAN_I TeSysU_IO_SCAN_O	HMI_MOTOR_TU_m	PUMP_PARAM
TeSys U on EPI2145	MOT_ EPI2145_2D	Stb_EPI2145_IO_SCAN_I Stb_EPI2145_IO_SCAN_O	HMI_MOTOR_EPI_2D	PUMP_EPI_PARAM



PLC and motor control device implementation

TeSys U controler

This section describes three references of TeSys_U control unit (Standard, Advanced, Multi-function), connected between STB main rack and STB extension rack.
An Ethernet IO scanning service is used to periodically exchange data between the PLC and the motor starter.
For TeSys_U behind an Advantys STB island, the STB island I/O mapping is used.



The table below details the interface between the starter and the DFB into PLC application.

TeSys U		UNITY PLC		
Register	Description	Located @ %MW	Variable (DDT)	DFB pin
455	State register	%MWx	TeSysUs_IO_SCAN_I (1) TeSysUa_IO_SCAN_I (2) TeSysUm_IO_SCAN_I	FbStat FbStat FbStat FbStat
458	I/O status register	%MWx+1		
461 (1) (2)	Warning register	%MWx+2		
457 (2)	Mechanical & power supply status register	%MWx+3		
704	Control Register	%MWy	TeSysU_IO_SCAN_O	QCtrl
703	Control of communication module	%MWy+1	TeSysU_IO_SCAN_O	
700	Output control	%MWy+2	TeSysU_IO_SCAN_O	

Note: The specific features relating to Advanced unit control are marked (1) and relating to Multi-function unit control are marked (2)
x=first input register, y=first output register (for more details see TeSys U configuration example in chapter 4)

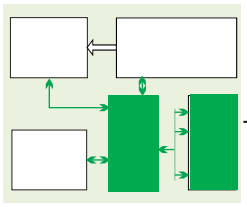
TeSys U controler DFB

DFB behaviour is compliant with the previous description (see page25),

Pin	-Interface	Type		MOT_TeSysU		Type	-Interface
1	Variable (IO_PLC)	Bool		Local	FbAuto	Bool	Variable (DDT)
2	Variable (IO_PLC)	Bool		Remote	FbManual	Bool	Variable (DDT)
					FbLocal	Bool	Variable (DDT)
					FbOff	Bool	Variable (DDT)
6	Variable (DDT)	Bool		Arun	FbRun	Bool	Variable (DDT)
7	Variable (DDT)	Bool		Lock	Locked	Bool	Variable (DDT)
8	Variable (DDT)	Bool		ExtErr	Error	Bool	Variable (DDT)
					Warning	Bool	Variable (DDT)
					Ave_Cur	Int	Variable (DDT)
13	Variable (IO_PLC)	Int*		FbStat	QCtrl	Int*	Variable (IO_PLC)
16	Variable (DDT)	Pump_Param		Param			
17	(HMI)	HMI_MOTOR_x		HMI	HMI		

(*) These variables are from the TeSysU_IO_SCAN structure (see description in the table below).

Input description	
FbStat	TeSys_U status words.
Output description	
Av_Cur (1), (2)	Average motor current in % of FLA.
QCrtl	Starter control word, sent to the starter via I/O scanning
In-output description	
(HMI)	Structure with command from HMI or SCADA and feedback to HMI or SCADA



PLC and motor control device implementation

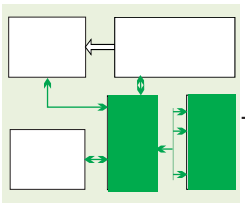
TeSys U controler DFB (cont'd)

HMI_MOTOR_TU_x (x=s for Standard, x=a for Advanced, x=m for Multi-function)		
Auto_Man	Bool	Command to set the block in Auto or Manu
Run_Stop	Bool	Command to run or stop the pump
Clear_Fault	Bool	Command to acknowledge internal and external errors indicated at the output Error. Acknowledgement is done with a rising edge.
Clear_Warning		Command to acknowledge starter warning. Acknowledgement is done with a rising edge.
Meas_1 (1),(2)	Int	Average motor current in % of FLA.
Nb_start	Int	Number of starts performed in the last 24 hours, displayed on HMI (calculated value)
Min_time_stop	Int	Minimum time between stop and start
Discrepancy_time	Int	Maximum time between and order and the right feedback
Time_to_start	Dint	Time before the motor restart, calculated when Param_Min_time_stop<>0.
Sts_Auto	Bool	Automatic mode is activated. The process sequence manages the motor
Sts_Manual	Bool	Manual mode is activated. The commands come from HMI or SCADA.
Sts_Local	Bool	Local mode is activated. The commands are hardwired.
Sts_Off	Bool	The TeSys is powered off by the hardwired.
Sts_Discrepancy_Err	Bool	Discrepancy error. The discrepancy control is activated (Discrepancy_time<>0).
Sts_Error	Bool	Indicates that the operation is blocked by an internal or external (Input Err) error, which is not acknowledged.
Sts_TimerProtect	Bool	Number start control is activated (Nb_start_day <>0) and the max starts number is reached.
Sts_Ready	Bool	TeSys-U ready to switch on
Sts_Run	Bool	The motor is running (Status from starter)
Sts_ExtErr	Bool	External error from the process or the system
Sts_Tsys_Fault	Bool	TeSys U all fault
Sts_Tsys_Rst_Auth	Bool	Reset TeSys fault authorized
Sts_Tsys_Warning	Bool	TeSys U all fault
W_Thermal (1), (2)	Bool	Thermal warning
W_Module (1), (2)	Bool	Module warning
Sts_Butt_On (2)	Bool	Button ON
Sts_Butt_On (2)	Bool	Button TRIP
Sts_Tsys_Closed (2)	Bool	Poles closed
Name	String[10]	Motor name & location

Note: The specific features relating to Advanced unit control are marked (1) and relating to Multi-function unit control are marked (2).

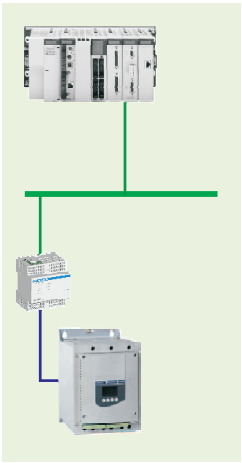
The table below details the interface between the starter and the DFB into PLC application I/O Interface, Inputs and output words are exchanged according to the table below

Type	Number	Comment
DI	2	Operating mode selector switch in "Remote" position.
		Operating mode selector switch in "Local" position.
DO	2	Auto mode indicator light
		Fault light
AI	-	Spare
AO	-	Spare



PLC and motor control device implementation

Soft starter ATS48



The ATS48 is connected to an Ethernet network via an ETG100 gateway.
Communication is transparent between the Ethernet and Modbus serial line therefore the I/O scanning service can directly access the ATS48 data.
The table below details the interface between the starter and the DFB into PLC application.

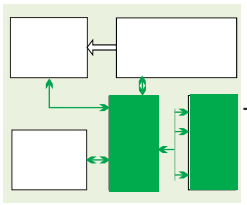
ATS48			UNITY PLC		
Register	Item	Description	Located @ %MW	Variable (DDT)	**DFB pin**
458	ETA	State register	%MWx	ATS48_IO_SCAN_I.Fb_Stat	FbStat
459	ETI	State register extended	%MWx+1	ATS48_IO_SCAN_I.Fb_Stat	FbStat
460	ETI2	State register extended	%MWx+2	ATS48_IO_SCAN_I.Fb_Stat	FbStat
400	CMD	Command register	%MWx+10	ATS48_IO_SCAN_O	QCtrl
4062	LCR	Motor curent	%MWy	ATS48_IO_SCAN_IM	Meas
4063	LTR	Motor torque	%MWy+1	ATS48_IO_SCAN_IM	Meas
4064	THR	Motor thermal state	%MWy+2	ATS48_IO_SCAN_IM	Meas
4065	PHE	Phase rotation direction	%MWy+3	ATS48_IO_SCAN_IM	Meas

Notes: x = First input register, y =First output register (for more details, see ATS48 configuration example in chapter 4)

ATS48 DFB

Pin	-Interface	Type	MOT_ATS48		Type	-Interface
1	Variable (IO_PLC)	Bool	Local	FbAuto	Bool	Variable (DDT)
2	Variable (IO_PLC)	Bool	Remote	FbManual	Bool	Variable (DDT)
				FbLocal	Bool	Variable (DDT)
				FbOff	Bool	Variable (DDT)
6	Variable (DDT)	Bool	Arun	FbRun	Bool	Variable (DDT)
8	Variable (DDT)	Bool	Lock	Locked	Bool	Variable (DDT)
9	Variable (DDT)	Bool	ExtErr	Error	Bool	Variable (DDT)
12	Variable (IO_PLC)	Int*	FbStat	QCtrl	Int*	Variable (IO_PLC)
14	Variable (IO_PLC)	Array*	Meas			
16	Variable (DDT)	Pump_Param	Param			
17	(HMI)	HMI_MOTOR_B	HMI	HMI		

(*) These variables are from the ATS48_IO_SCAN structure (see description in table below)
This DFB is compliant with the description in page 27 design chapter, therefore only specific pins are detailed.



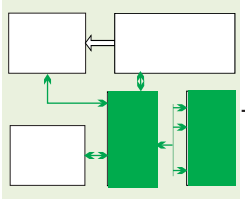
PLC and motor control device implementation

ATS 48 DFB

(cont'd)

Input description	
FbStat	Starter status word. Feedback signal from I/O scanning
ATS_Meas	Starter measures. Feedback signal from I/O scanning
	The measure variable depends on the first register address
Output description	
QCrtl	Starter control word, sent to the starter via I/O scanning
In-output description	
(HMI)	Structure with command from HMI or SCADA and feedback to HMI or SCADA

Used structures		
HMI_MOTOR_B		
Auto_Man	Bool	Command to set the block in Auto or Manu
Run_Stop	Bool	Command to run or stop the pump
Rst_Fault	Bool	Command to acknowledge internal and external errors indicated at the output Error. Acknowledgement is done with a rising edge.
Speed_Output	Int	Output speed from the ATS.
Meas_1	Int	Object to be configured in I/O scanner and to be displayed in HMI (eg: current, power consumption,…)
Meas_2	Int	Object to be configured in I/O scanner and to be displayed in HMI (eg: current, power consumption,…)
Meas_3	Int	Object to be configured in I/O scanner and to be displayed in HMI (eg: current, power consumption,…)
Meas_4	Int	Object to be configured in I/O scanner and to be displayed in HMI (eg: current, power consumption,…)
Nb_start	Int	Number of starts performed in the last 24 hours, displayed on HMI (calculated value)
Min_time_stop	Int	Minimum time between stop and start
Discrepancy_time	Int	Maximum time between and order and the right feedback
Time_to_start	Dint	Time before themotor restart, calculated when Param_Min_time_stop<>0..
Sts_Auto	Bool	Automatic mode is activated. The process sequence manages the motor
Sts_Manual	Bool	Manual mode is activated. The commands come from HMI or SCADA.
Sts_Local	Bool	Local mode is activated. The commands are hardwired.
Sts_Off	Bool	The ATS is powered off by the hardwired.
Sts_Locked	Bool	Indicates that the operation is blocked by an interlock (input Lock).
Sts_Error	Bool	Indicates that the operation is blocked by an internal or external (Input Err) error, which is not acknowledged.
Sts_TimerProtect	Bool	Number start control is activated (Nb_start_day <>0) and the max starts number is reached.
Sts_ATS_Ready	Bool	Starter ready to switch on
Sts_ATS_Run	Bool	The motor is running (Status from starter)
Sts_ATS_PTC	Bool	Motor monitoring by PTC probe
Sts_ATS_Estop	Bool	ATS emergency stop activated
Sts_ExtErr	Bool	External error from the process or the system
Sts_NoVoltage_Err	Bool	No voltage in ATS
Sts_ATS_Err	Bool	ATS in error
Sts_Discrepancy_Err	Bool	Discrepancy error. The discrepancy control is activated (Discrepancy_time<>0).
Name	String [10]	Motor name & location



PLC and motor control device implementation

ATS 48 DFB(cont'd)

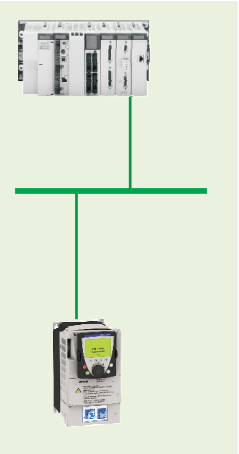
Functional description

DFB behaviour is compliant with the previous description (see page 27),

Type	Number	Comment
DI	4	Operating mode selector switch in "Remote" position.
		Operating mode selector switch in "Local" position.
		Circuit breaker open
		ATS fault
DO	2	Motor line contactor
		Auto mode pilot light
AI	-	Spare
AO	-	Spare

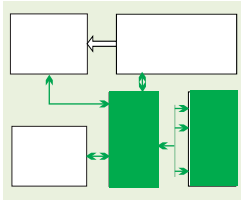
Variable speed drive ATV61

The ATV61 is connected on Ethernet Ethernet IO scanning service is used to periodically exchange data between the PLC and variable speed drive.
Input and output words are exchanged according to the table below.
ATV61 allows an interchange of 10 words In and 10 words Out. In our DFB, to manage the ATV61 starter, 5 input words are read and 3 output are written. Additional words can be read and written by a simple modification of the I/O scanning configuration.
(see chapter Configuration).



The table below details the interface between the starter and the DFB into PLC application.

ATV61			UNITY PLC			
Register	**Item**	**Description**	**Located @ %MW**	**Variable (DDT)**	**DFB pin**	Comment
	Reserved		%MWx			Word reserved by the system
3201	ETA	State register	%MWx+1	ATV61_IO_SCAN.Fb_stat	FbStat	
8604	RFRD	Actual speed value	%MWx+2	ATV61_IO_SCAN.Fb_RPM	FbRPM	
3204	LCR	Motor curent	%MWx+3	ATV61_IO_SCAN.Measure	Meas	Free of configuration
3218	IPR	Input power	%MWx+4	ATV61_IO_SCAN.Measure	Meas	Free of configuration
8501	CMD	Command register	%MWx+5	ATV61_IO_SCAN.Q_Ctrl	QCtrl	
8602	LFRD	Target velocity	%MWx+6	ATV61_IO_SCAN.Q_RPM	QRPM	



PLC and motor control device implementation

ATV61 DFB (cont'd)

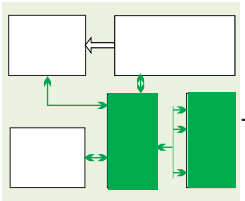
This DFB is compliant with the description in page 27 design chapter, therefore only specific pins are detailed.

Pin	-Interface	Type	MOT_ATV61		Type	-Interface
1	Variable (IO_PLC)	Bool	Local	FbAuto	Bool	Variable (DDT)
2	Variable (IO_PLC)	Bool	Remote	FbManual	Bool	Variable (DDT)
				FbLocal	Bool	Variable (DDT)
				FbOff	Bool	Variable (DDT)
6	Variable (DDT)	Bool	Arun	FbRun	Bool	Variable (DDT)
7	Variable (DDT)	Int	RPM			
8	Variable (DDT)	Bool	Lock	Locked	Bool	Variable (DDT)
9	Variable (DDT)	Bool	ExtErr	Error	Bool	Variable (DDT)
12	Variable (IO_PLC)	Int*	FbStat	QCtrl	Int*	Variable (IO_PLC)
13	Variable (IO_PLC)	Int*	FbRPM	QRPM	Int*	Variable (IO_PLC)
14	Variable (IO_PLC)	Array*	Meas			
16	Variable (DDT)	Pump_Param	Param			
17	(HMI)	HMI_MOTOR_A	HMI	HMI		

(*) These variables are from the ATV61_IO_SCAN structure (see description in relationships table below).

The inputs/outputs and the functions are quickly described. The function block is available from Schneider End user solution Web Site.

Input description	
RPM	Setpoint for motor speed in Manual mode.
FbStat	Starter status word. Feedback signal from I/O scanning
FbRPM	ATV output speed. Feedback signal from I/O scanning
ATV_Meas	Starter measures. Feedback signal from I/O scanning
	The type of measure variable is defined in the ATV I/O scanner.
Output description	
QCrtl	Starter control word, sent to the starter via I/O scanning
QRPM	ATV setpoint speed, sent to ATV via I/O scanning
In-output description	
(HMI)	Structure with command from HMI or SCADA and feedback to HMI or SCADA



PLC and motor control device implementation

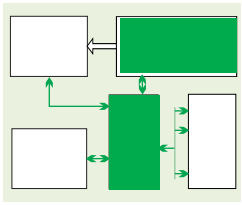
ATV61 DFB (cont'd)

Used structures		
HMI_MOTOR_A		
Auto_Man	Bool	Command to set the block in Auto or Manu
Run_Stop	Bool	Command to run or stop the pump
Rst_Fault	Bool	Command to acknowledge internal and external errors indicated at the output Error. Acknowledgement is done with a rising edge.
Speed_Setpoint	Int	Setpoint for motor speed in Manual mode.
Speed_Output	Int	Output speed from the ATV
Meas_1	Int	Object to be configured in the I/O scanner and to be displayed in the HMI (eg: current, power consumption,…)
Meas_2	Int	Object to be configured in the I/O scanner and to be displayed in the HMI (eg: current, power consumption,…)
Nb_start	Int	Number of starts performed in the last 24 hours, displayed on HMI (calculated value)
Min_time_stop	Int	Miminum time between stop and start
Discrepancy_time	Int	Maximum time between a command and the right feedback
Time_to_start	Dint	Time before the next start, calculated when Nb_start reaches the limit, displayed on HMI.
Sts_Auto	Bool	Automatic mode is activated. The process sequence manages the motor
Sts_Manual	Bool	Manual mode is activated. The commands come from HMI or SCADA.
Sts_Local	Bool	Local mode is activated. The commands are hardwired.
Sts_Off	Bool	The ATV is powered off by the hardwire.
Sts_Locked	Bool	Indicates that the operation is blocked by an interlock (input Lock).
Sts_Error	Bool	Indicates that the operation is blocked by an internal or external (Input Err) error, which is not acknowledged.
Sts_TimerProtect	Bool	Number start control is activated (Nb_start_day <>0) and the max starts number is reached.
Sts_ATV_Ready	Bool	Starter ready to switch on
Sts_ATV_Run	Bool	The motor is running (Status from starter)
Sts_ATV_Fwd	Bool	Motor running in positive direction
Sts_ATV_Bwd	Bool	Motor running in negative direction
Sts_ATV_Estop	Bool	ATV emergency stop activated
Sts_ExtErr	Bool	External error from the process or the system
Sts_NoVoltage_Err	Bool	No voltage in ATV
Sts_ATV_Err	Bool	ATV in error
Sts_Discrepancy_Err	Bool	Discrepancy error. The discrepancy control is activated (Discrepancy_time<>0).
Name	String[10]	Motor name & location

Functional description
The DFB behaviour is compliant with the previous description (see page 27), except in the following case. If the operation mode is changed from automatic mode to manual mode, the motor continues at the same speed. If the operation mode is changed from manual mode to automatic mode, the motor will follow the commands from the process.

I/O interface associated to ATV61 management needs additional input and output data to control the operating mode, circuit breaker and contactor state. These I/Os are located in Advantys STB.

Type	Number	Comment
DI	4	Operating mode selector switch in "Remote" position.
		Operating mode selector switch in "Local" position.
		Circuit breaker open
		Drive fault
DO	2	Motor line contactor
		Auto mode pilot light
AI	-	Spare
AO	-	Spare



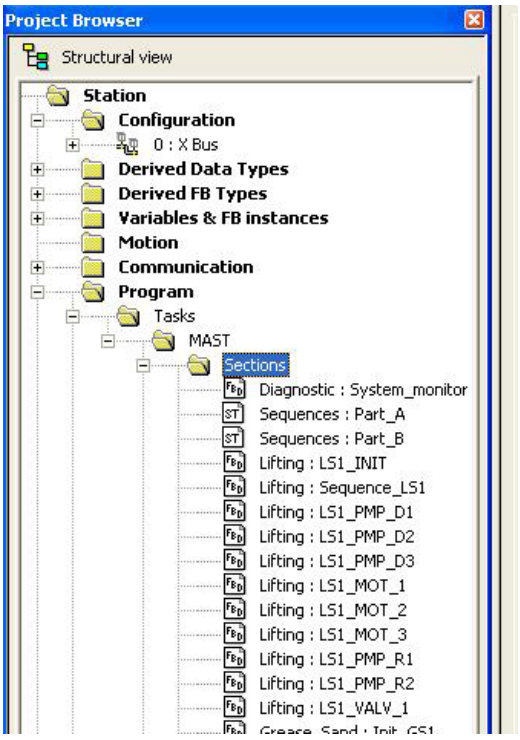
Unity program implementation

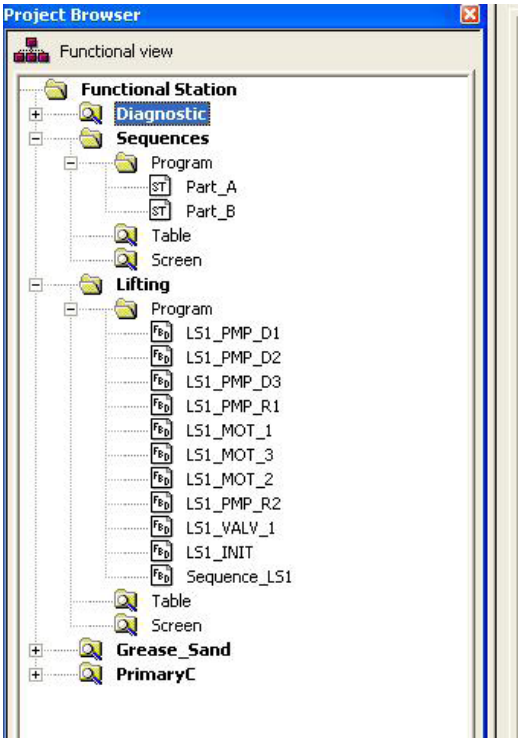
UNITY program structuring

The UNITY program comprises several sections, some of them with a transversal role such as PLC system monitoring or the process sequence.
PLC system monitor the system state (alarms, status, communication...).

Process sequences coordinate the process functions.

In Unity, the program can be represented in two ways:
- **Using a structural view,** directly related to the PLC application
- **Using a functional view,** which allows greater readability of the process





- **Functional view**

A good way to structure a program is by defining functional modules. This method has multiple advantages:
- The readability of the program, useful for maintenance or development
- The ability to duplicate a process functional unit easily by export/import

Variables naming rules

The following Tag naming rules is defined before developing the application:

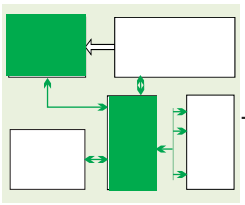
XXX_YYY_ZZZ

XXX: Variable group.
Possible values:

(HMI)	: variables exchanged with SCADA/HMI
IO	: I/O variables used to control the devices (I/O scanner)
I	: Physical input
Q	: Physical output

For variables used internally in PLC code (no exchange on I/O, networks, or buses), “XXX_” is omitted.

YYY: Identification of the functional unit to which the device belongs.
ZZZ: Identification of the device type such as pump, motor, valve.

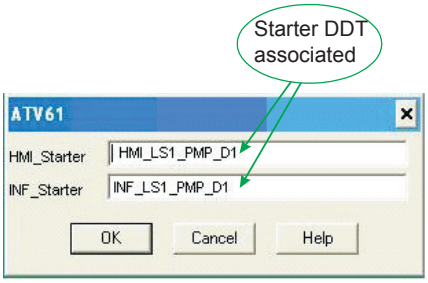


SCADA implementation

SCADA principle

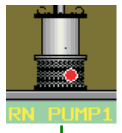
Genies and Super Genies objects are defined to design the motor control application. They use structured variables that come from XVM files generated by Unity.
For each type of graphical object (pump, motor,…) a genie is created. This genie can be pasted from a genie dialog box and added to the graphics page. When the genie is pasted in the graphical page subsequently, a popup window is activated to substitute the tags used in the Super Genie called by the genie. Each super genie is associated to a Cicode function; this function :

- substitutes tags in the Super Genie
- opens the Super Genie

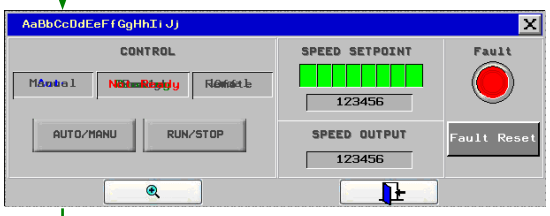


SCADA example

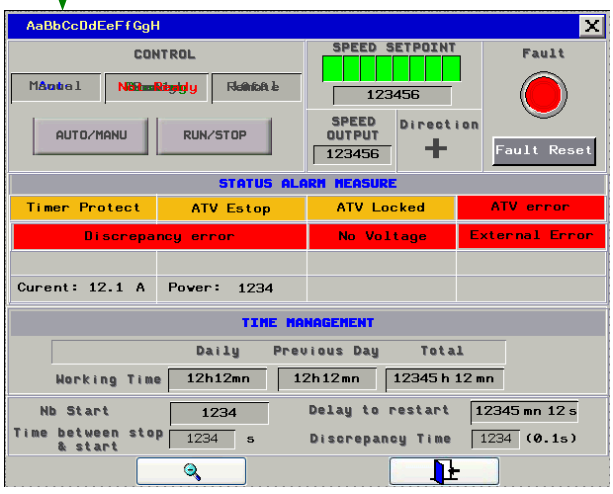
For pump or motor management a Genie is linked to two Super Genies (figure below). Clicking on a Genie pump object opens a first popup (Super Genie) which provides a first level of diagnostic information. In order to have additional diagnostic information a “detail” button has to be used to open a second popup (Super Genie).

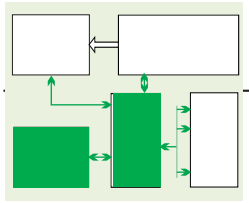


On Click
Cicode Open Small_HMI_a and pass variables



On Click
Cicode Open Big_HMI_a and pass variables





HMI implementation

Principle

Vijeo Designer 4.6 can not use the data structure from Unity, therefore the following recommendation are proposed.

Example

Example of HMI data mapping
Two ATV61 manage two separate pumps. A Unity structured data type HMI-ATV is defined to create the variables used for HMI exchanges.
In the example, two instances are created and link to the two ATV61 drives.
The parameters from the first ATV61 are assgined to LS1_PMP_D1 (type HMI_ATV) mapped in %MW100 on Unity side.
The second ATV is assigned to LS1_PMP_D2 (type HMI_ATV) mapped on %MW140. The size of an HMI_ATV variable is 40 words, this value will be used to manage the index on the HMI side.

In Vijeo Designer, a similar HMI_ATV data type is used.
3 types of variables are used Boolean, Integer, String[10]. Therefore 3 indexes are needed:
_ Index_Bool
_ Index_Int
_ Index_String

At pump popup call, the index values are updated. An example of index usage is provided below:

Calculation of index: Index =Gap * Type variable.
Variable Integer, type = 1 => Index_Int =40 * 1 = 40
Variable Boolean, type =16 => Index_Bool = 40 * 16 = 640
Variable String[10], type = 1/5 =>Index String = 40/5= 8

UNITY

HMI_ATV

Type

Aut_Man
Run_Stop
Clear_Flt
Speed_Setpoint
Name

HMI_LS1_PMP_D1 (type HMI_ATV) %MW100

Instance 1

Aut_Man	%MW100.0
Run_Stop	%MW100.8
Clear_Flt	%MW101.0
Speed_Setpoint	%MW102
Name	%MW119

HMI_LS1_PMP_D1 (type HMI_ATV) %MW140

Instance 2

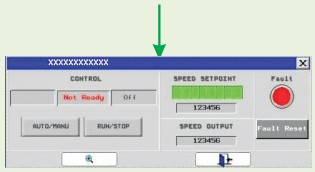
Aut_Man	%MW140.0
Run_Stop	%MW140.8
Clear_Flt	%MW141
Speed_Setpoint	%MW142
Name	%MW159

VIJEO DESIGNER

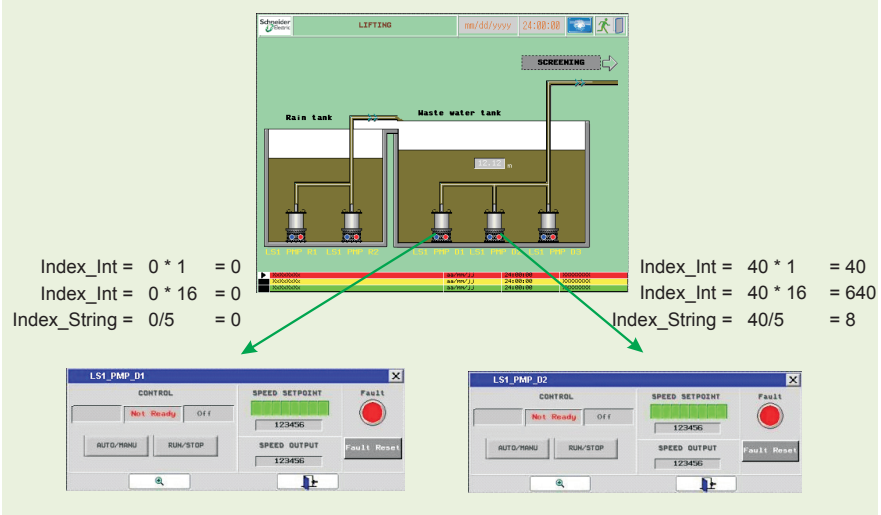
HMI_ATV

Aut_Man	%MW100.0 [Index_bit]
Run_Stop	%MW100.8 [Index_bit]
Clear_Flt	%MW101.0 [Index_bit]
Name	%MW1119[Index_String]
Speed_Setpoint	%MW1102 [Index_word]

Variables used by the popup



Popup(s) and Index



6-Operation

Table of contents

Introduction	56
Running a Process Control and Diagnostics	56
Architecture	56
Example	57

Introduction

Purpose

In this chapter, available operations on process control and related motor control diagnosis are described.

Process Control and Diagnostics

Process control

Process control requires information to monitor the different sections of the process and the functions related to motor control.
It is necessary to be able to give commands and settings to the pumps and motors associated with the process equipments. It is important to be able to manage event histories (cycles, operating time...) and to manage measured values (levels, flows, power...).

Diagnosis of operating functions

In principle, four four levels of defects must be managed:
- Process defects such as flow problems (level, pressure) or problems with the quality of materials.
- Equipment defect related to a process section such as a conveyor, settler, or crusher.
- Motor or pump fault.
- Fault in motor control, such as the soft starter or the VSD.

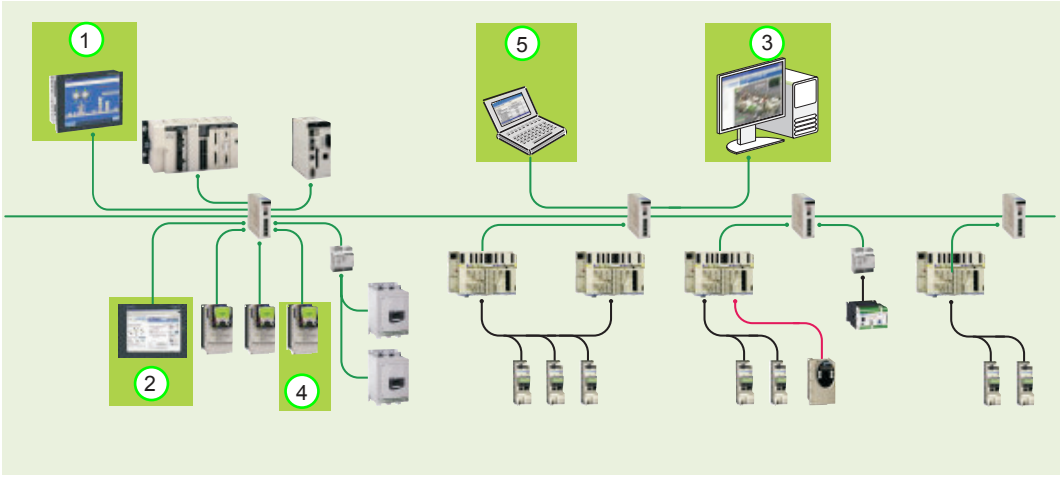
Alarms are classified into different levels to help diagnostics and reduce the potential downtime.

Architecture

Purpose

The actions necessary to manage these various levels of defects are performed by personnel who have various qualifications and use specific tools. Consequently, the presented architecture implements various diagnosis solutions.

Five interfaces are used for process control functions and/or diagnostic functions:
1 The SCADA Vijeo Citect which provides complete monitoring of the process and the ability to control in manual mode.
2 The HMI Magelis XBTGT provides a local monitoring of the process. But, it also provides a global vision of the plant.
3 The Web diagnostic services provides system diagnosis during the maintenance phase. Products, such as PLCs, ATV61, Advantys STB and ETG100 gateway are embedded with pre-loaded diagnostic pages that can be visited from a standard web browser.
4 Buttons and indicators, provide quick display of the equipment status and also permit local command operations on motors and pumps.
5 Dedicated software tools such as Unity or PowerSuite which allow diagnosis of equipment and processes, particularly in the development and implementation phase of the process.

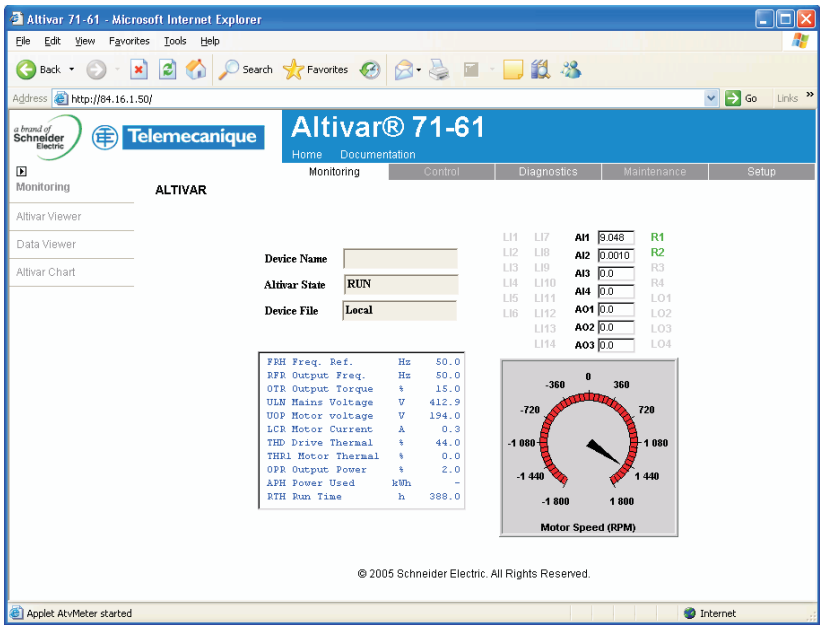


Example

Access to an ATV61

In this type of architecture, we can access a starter in several ways.
From the SCADA Vijeo Citect (1), zoom to the functional unit and click on the corresponding motor starter.

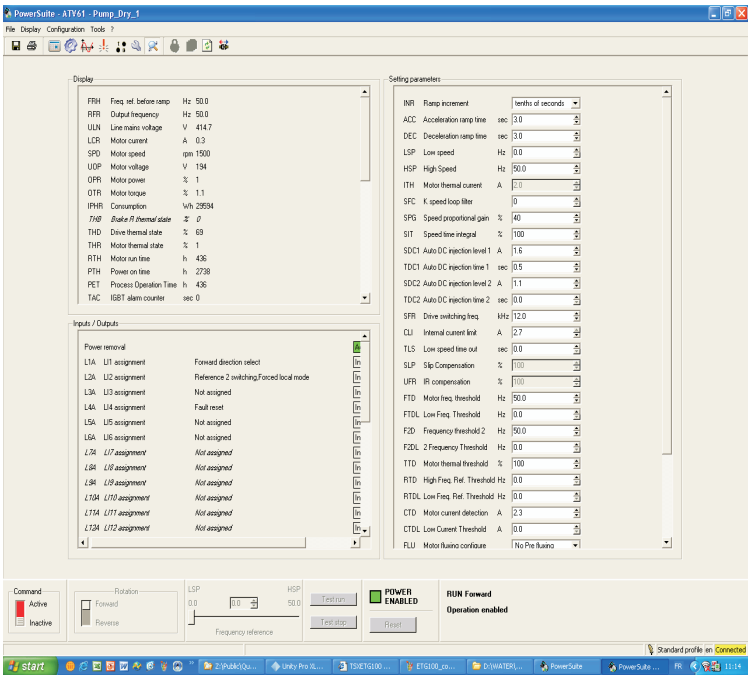
From a web client station (3) with Internet Explorer, type in the IP address of the motor starter.
Note: Default login name "USER" and default password "USER".



Example

Example (cont'd)

From a PC (5), we can access a starter using the PowerSuite tool.
Type in the IP address of the motor starter.



From the graphical terminal on the starter (4).

Water application example

Table of contents

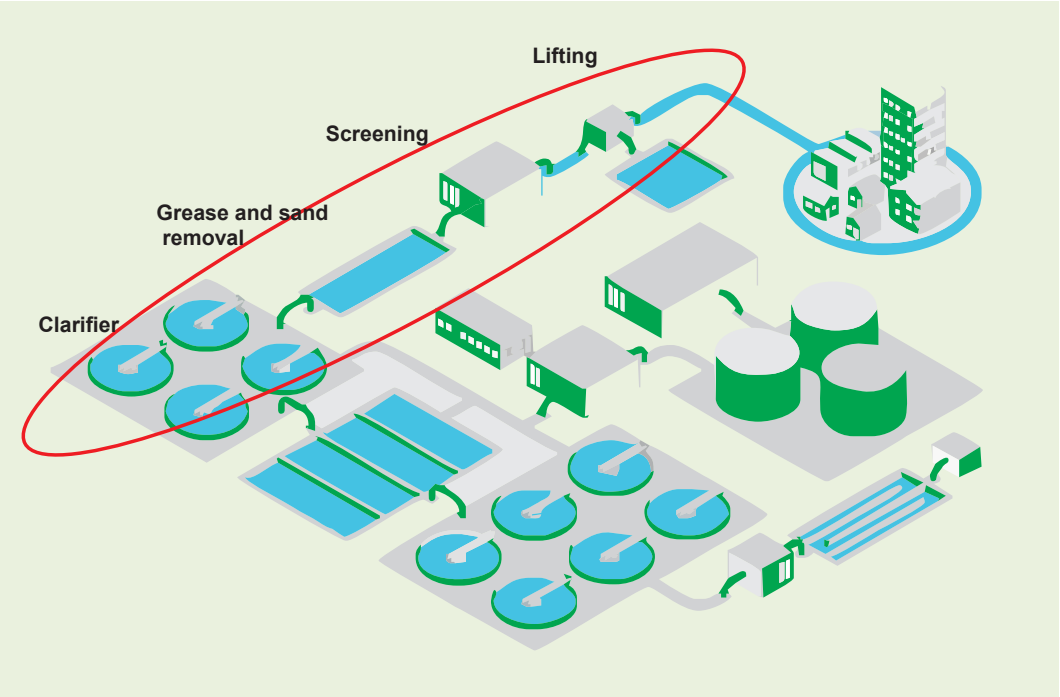
Objectives	60
Selection phase	61
Design phase	62
Configuration phase	70
Implementation phase	70
Operation phase	73
Components list	76

Objective

Presentation

The purpose of this chapter is to provide a real example of a process control application using the recommendations in the previous chapters.

A waste water treatment plant is used to illustrate how to realize a pump and motor application. The first part of the process is implemented from the lifting to the primary clarifier using the solution platform introduced before.



Process step

Each process step has specific constraints in terms of pumping and motors.

Process step	Short description	Pumps & motors
Lifting	To ensure the efficient gravity flow in the plant, wastewater has to be lifted up.	-Redundant pumps to feed in water -Redundant pumps for rain water tank.
Screening	To remove the most voluminous trash in order to facilitate the next treatment phases.	-Motor for Grid management and Scraper -Motor for waste evacuation -Motor for waste compression
Sand & grease removal	To separate wastewater from grit and sand with the help of air. The air that is blown into the tank resolves the concentrated waste.	-Motor for scraper -Motor for the shield -Redundant pumps to extract the sand and grit
Primary clarifier	To remove settled and floating or suspended materials from the wastewater. Primary sludge is pumped out of the clarifier	- Motor for scraper - Motor for the shield

In the following parts of the document, the methodology described above is used, from the selection phase to the implementation and finally the operation phase.

Selection

Motor control device selection

The following table describes the selection criteria used to select the most appropriate motor control devices.

			DOL	Soft starter	VSD
Lifting	Feed in pumps	Redundant centrifugal pumps: Flow is controlled by changing the speed of rotation in order to better manage water flow variation.			ATV61 redun-dant
	Rain water pumps	Centrifugal pumps: Flow is controlled easily by valves on the pump discharge manifolds. The rain water tank allows easy management of flow variation.		ATS48	
Screening	Motor for scraper	Motor for moving device. Constant and low speed.	TeSys U		
	Motor for waste compression	Motor to control valve with 2 positions.	TeSys U		
	Motor for waste evacuation	Motor for moving device with no speed constraint.	TeSys U		
Sand & grease removal	Air compressor	The power required to manage an air compressor can be significant (>15KW). For this basin the requested pressure is constant.	TeSys T		
	Valve for sand outlet	Valve with 2 positions with no specific con-straint.	Contactor LC.D or F		
	Motor for scraper	Motor for moving device, with the possibility of optimising the function			ATV31
	Motor for the shield	Motor to control the shield. No specific con-straint linked to this equipment.	TeSys U		
	Pump to extract the sand	Redundant pumps to ensure sand extraction in the event of a fault	TeSys U LC.D or F		
Primary clarifier	Motor for scraper	Motor for moving device. Constant and low speed.	TeSys U		
	Motor for the shield	Motor to control the shield. No specific constraint linked to this equipment.	TeSys U		
	Valve for sand outlet	Valve with 2 positions with no specific constraint	Contactor LC.D or F		

Selecting architecture:

The criteria described in chapter 2 were applied to build the communication architecture.

Type of links	D.O.L. starter					Soft starter	VSD	
	Motor circuit breaker + Contactor LC.D or F	Starter controller TeSys U			Motor management system TeSys T	ATS48	ATV31	ATV61
		Standard control unit	Advanced control unit	Multifunction control unit				
Hard wired	Sand & grease removal							
Modbus SL					Sand & grease removal	Lifting		
CANopen								
Ethernet								Lifting
Advantys + Pre wiring		Primary clarifier						
Advantys extension bus		Screening	Screening	Screening				
Advantys CANopen							Sand & grease removal	

The architecture proposed is consistent in terms of the communication solution and information monitoring. It also shows different levels of architecture in term of cost and performance.

Design

Introduction

The water application architecture is designed using the solution platform introduced previously.
The different process steps are split in three cabinets:
- Main control cabinet
- Lifting and Screening cabinet
- Grease & sand removal and primary clarifier cabinet.

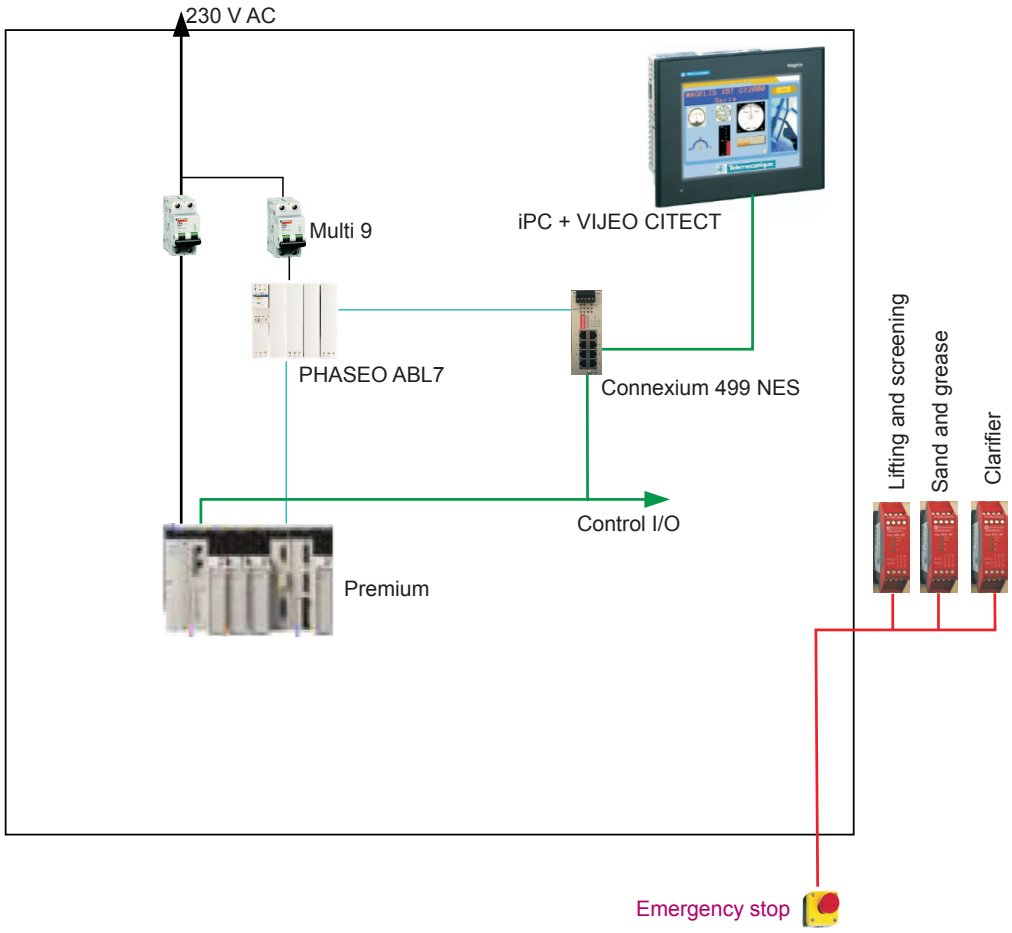
Main control cabinet

The main cabinet 1 contains the Modicon Premium P574634 PLC with the integrated Ethernet module.
This PLC manages the distributed I/O and motor control devices located in other cabinets.
The standalone SCADA application connected to the PLC (Vijeo Citect software) is performed on a Magelis iPC. This application can follow and control the global process as well as the control of each actuator.
It includes alarms, events, trends and history records.

Cabinet 1
Main control cabinet

Design

Main control cabinet



Design

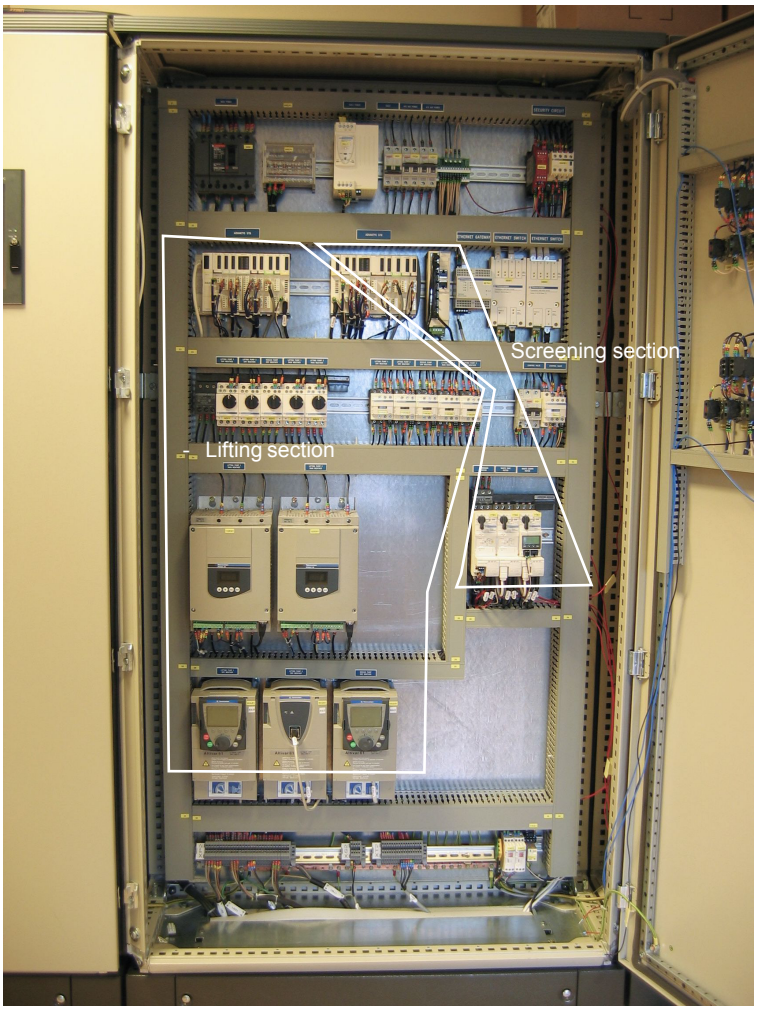
Lifting and screening

This cabinet integrates the lifting and screening sections.
- An Ethernet network connects each island section. Some products (VSD) are directly connected to Ethernet, the rest are managed across an Advantys STB I/O module.
- A HMI graphic terminal XBTGT is connected to Ethernet. It enables management of the local process and the process of other functions units (Sand & Grease and Primary Clarifier).

This architecture can be extended if required, a HMI can be added locally in each functional unit via the Ethernet connection.

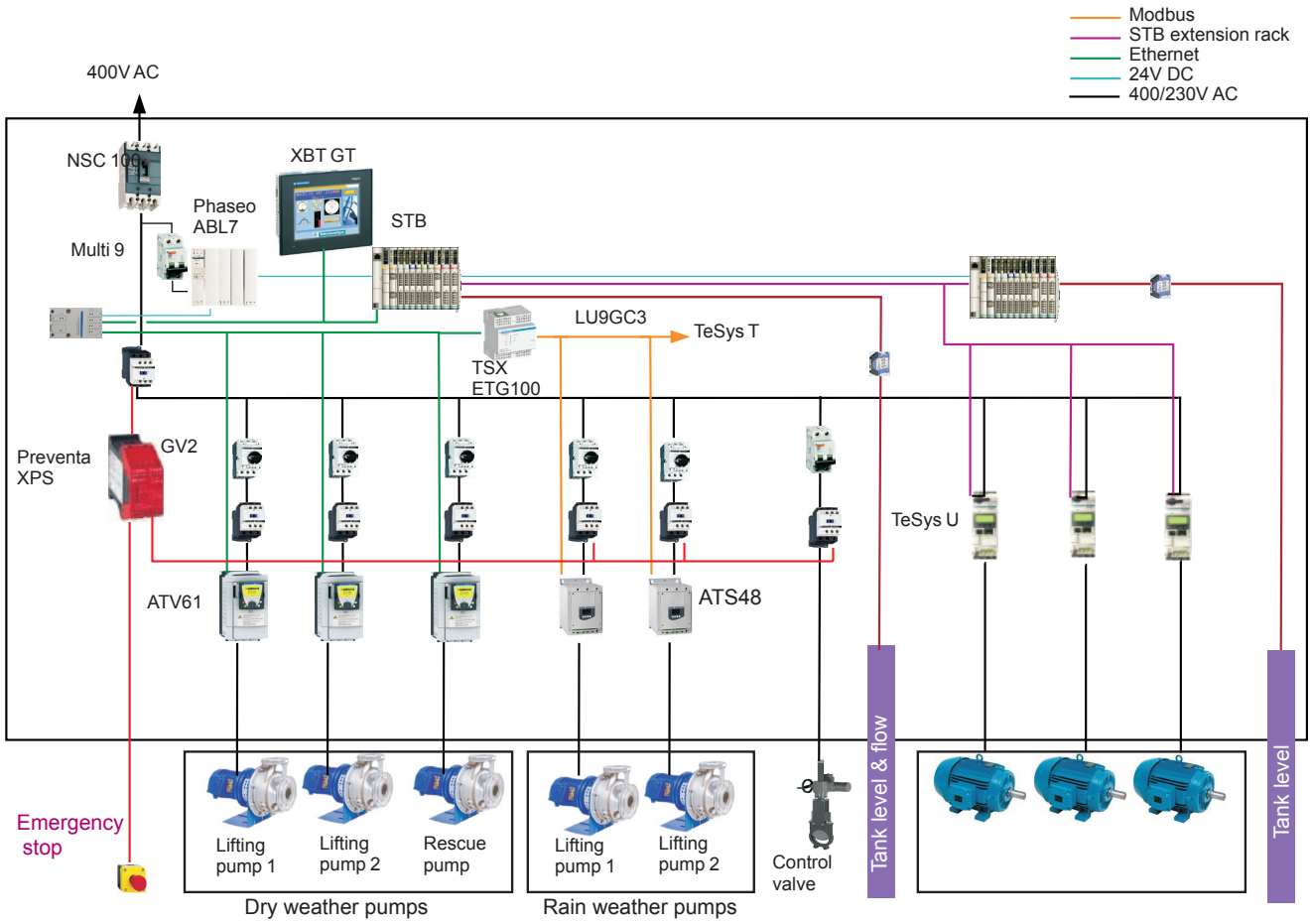
To illustrate the different applications, different types of motor starters are used.
□ For pump control in the lifting section:
- Dry weather pumps are controlled with variable speed drive Altivar 61 connected on Ethernet. Each VSD is associated with a contactor and a circuit breaker. Two pumps for normal range, and a third one in rescue.
- Rain water pumps are controlled with soft starter ATS48. They are connected to Ethernet through an Ethernet gateway to a Modbus serial line. Continuity of service is guaranteed.

□ For wet weather we have two complementary pumps are used to absorb the additional flow.



Design

Lifting and screening



Design

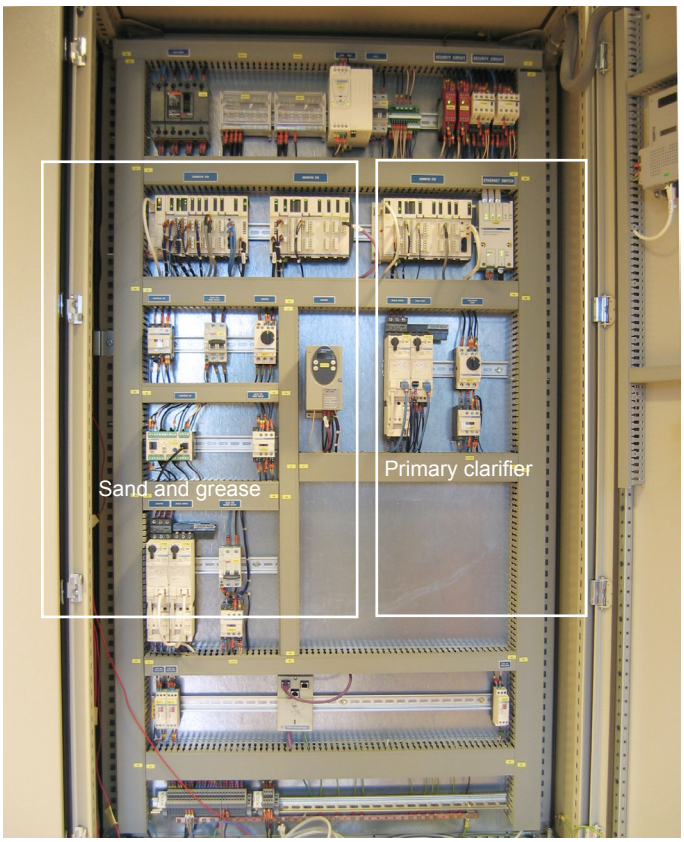
Sand and grease removal and primary clarifier

This cabinet has the same architecture with two Advantys STB automation islands
In this example, the proposed motor control solutions are:

■ **For grease & sand compressor**
High power motor requires specific parameter control like phase reversing, diagnostics...
A TeSys T multifunction relay is used.
The product is linked to the Ethernet across the Modbus gateway of cabinet 2.
All required parameters can be accessed via the Modbus protocol.
Contactor control is performed with hard wiring connected to the Advantys STB I/O modules.

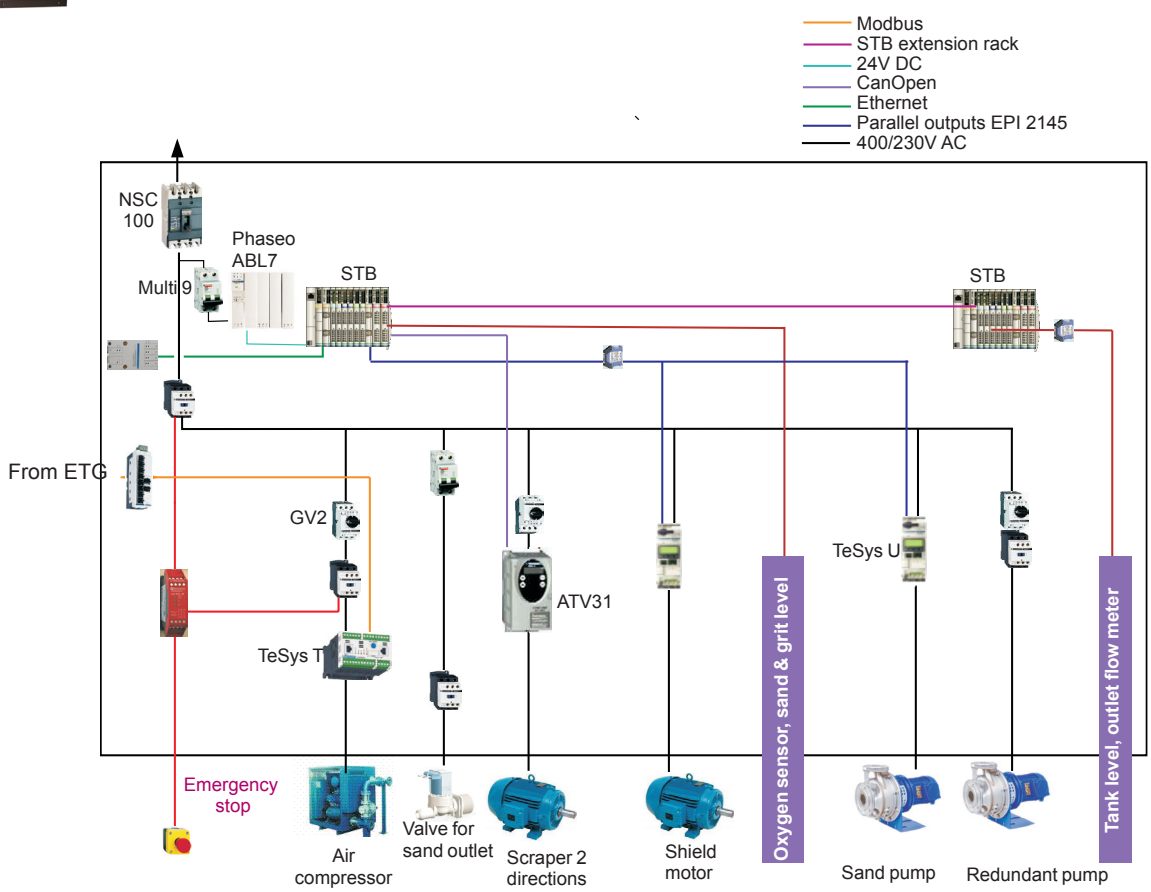
■ **For scrapper**
The reversing control of the motor uses an ATV31 drive connected on the CanOpen extension bus of the STB island.

■ **For water pumps**
A direct on line TeSys-U connected on Advantys STB EPI2145 module with pre-wired connections is used. A redundant pump is directly managed by a contactor connected to Advantys STB I/O.

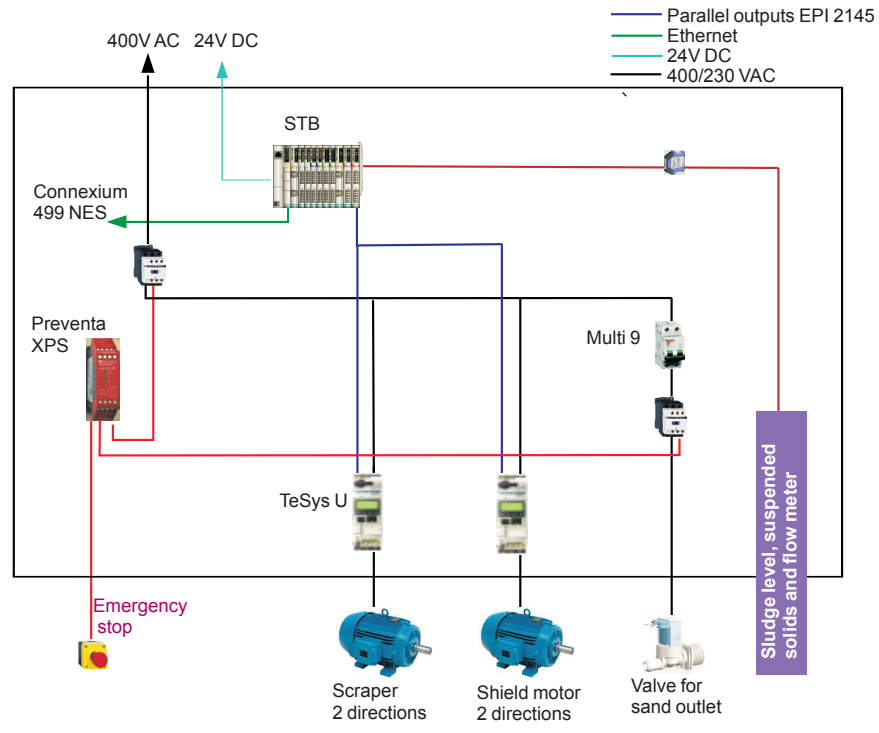


Design

Sand and grease



Primary clarifier



Design

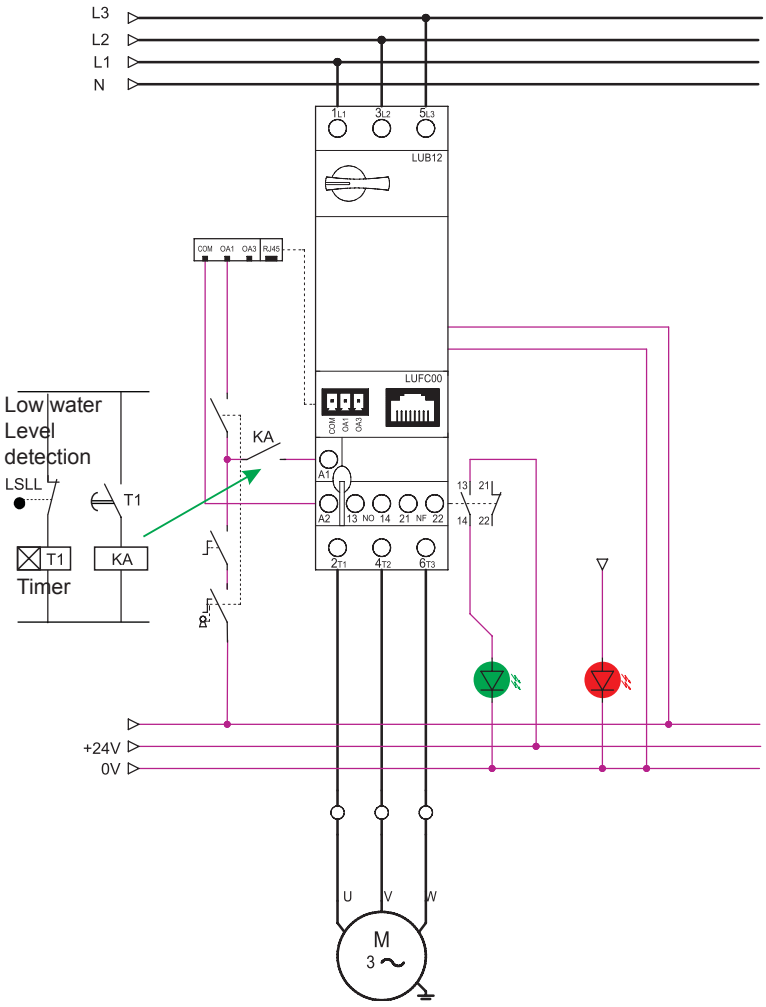
Specific pump functions

For applications like water treatment or pumping, some complementary functions are needed.
Diagrams of two typical functions are presented:

- **Running dry detection**
- **Detection of excessive or negative pressure**

Running dry detection

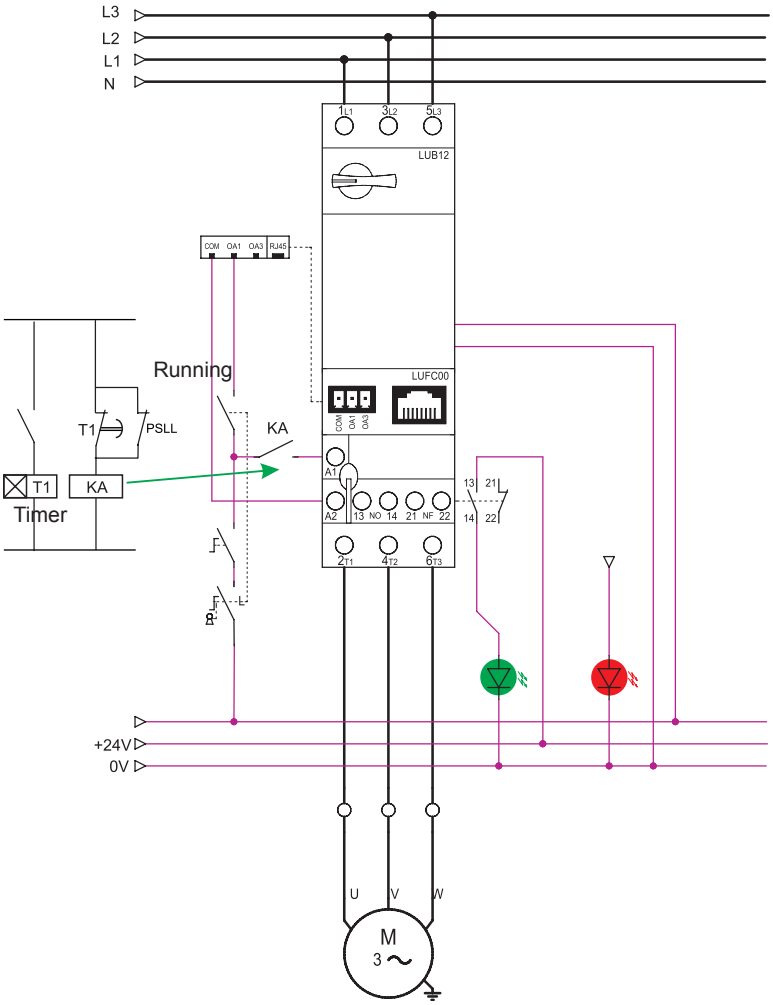
This function is used to manage low water level



Operation sequences :
1 Normal level = Tempo increase
2 Tempo ending = Start pump
3 Abnormal level = Stop pump
4 Normal level = Tempo increase
5 Tempo finished = Start pump
Note: Delaying compensates sensoroscillations

Design

Detection of excessive or negative pressure



Operation sequences:
1 Alarm pressure
2 Pump running= Tempo increase
3 Pressure OK
4 Pump running
5 Alarm pressure = Stop pump
Remark: Delaying allows pump start with pressure alarm

Configuration

Introduction

The configuration phase uses all elements described in chapter 4.

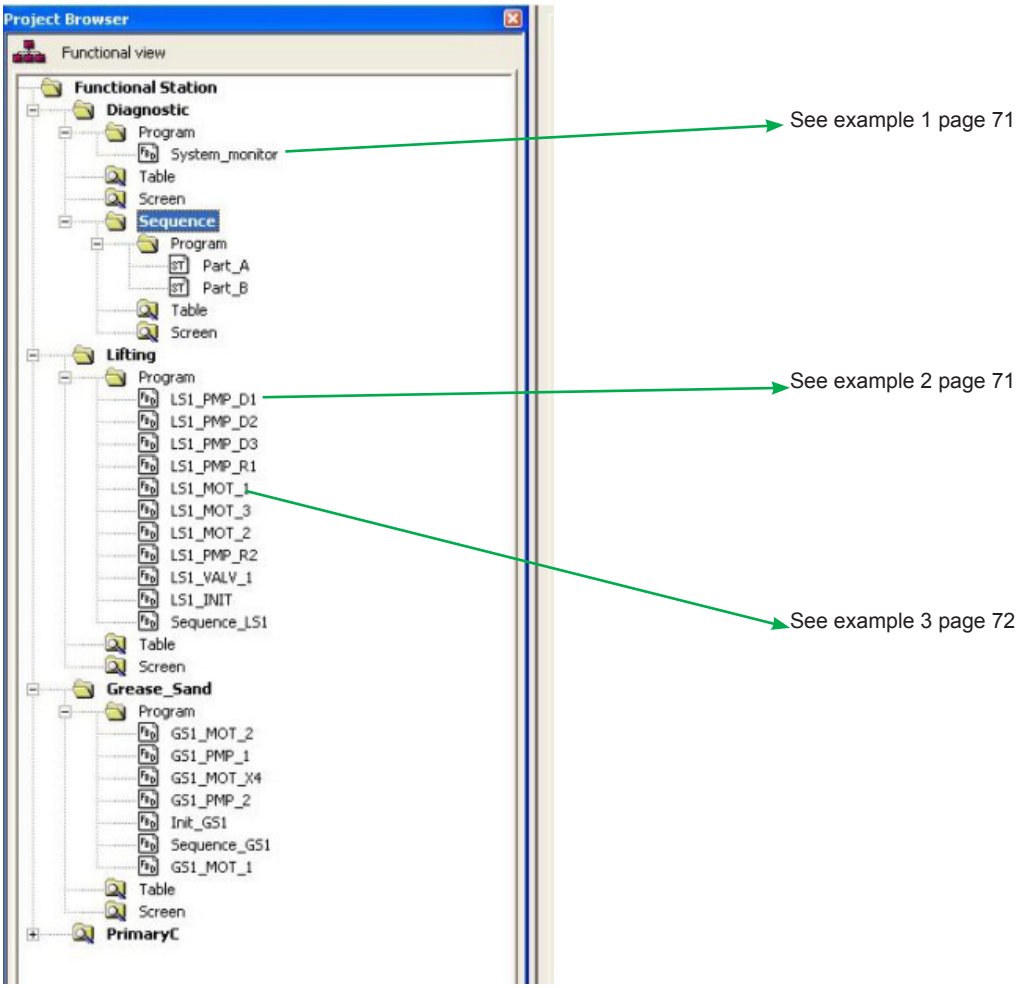
Implementation

Unity program

The UNITY program comprises several sections, some of witch perform a transversal role such as PLC system monitoring or the process sequence coordination.

■ **PLC system monitoring manages the system state** (alarms, status, communication...).

■ **Processes sequences coordinate the process functions, and link whole functional units:**
□ Init section: Coordinates the init sequence of all starters.
□ Sequence section: Manages the sequence process inside the functional unit. This section is linked to process section
□ Starter section: Allows direct management of the starters from the SCADA or HMI to starter working modes.
In these sections the DFBs are in accordance with the starter type being managed. In the application, a section is dedicated to one starter. In case of a significant number of motors, it is easier to put all motor function units together.

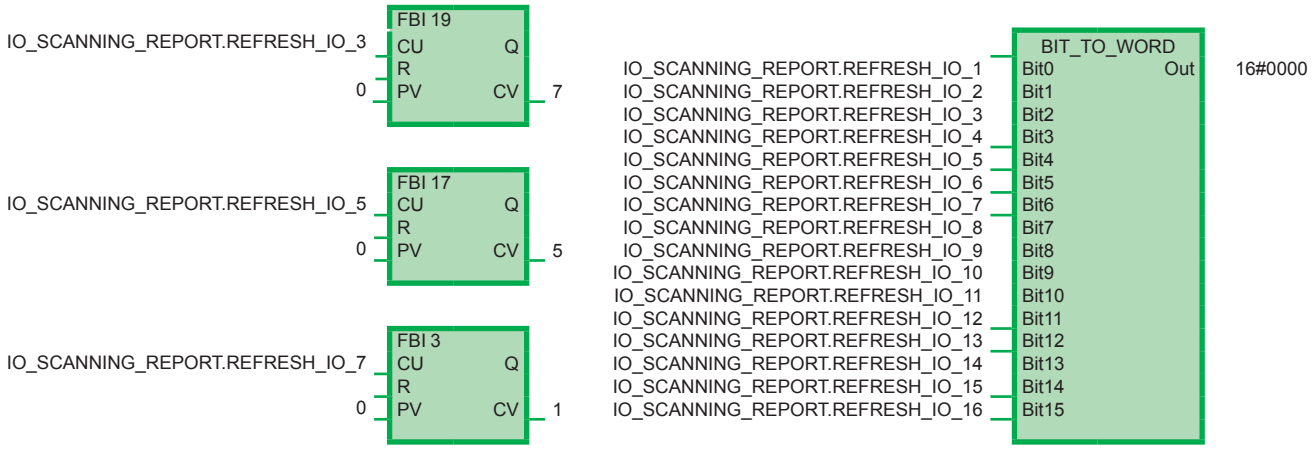


Implementation

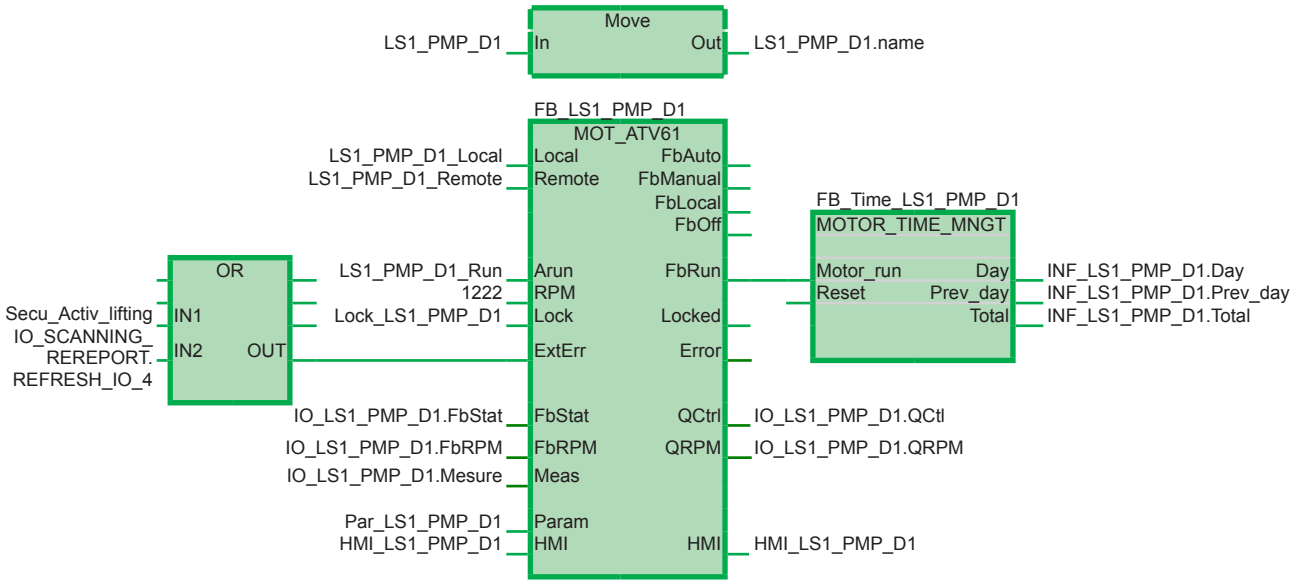
Unity program (cont'd)

1-Example of diagnostics for devices on the I/O scanner

In this section, equipement status is monitored, an IODDT "T_COM_ETHCOPRO" dedicated to Ethernet is used



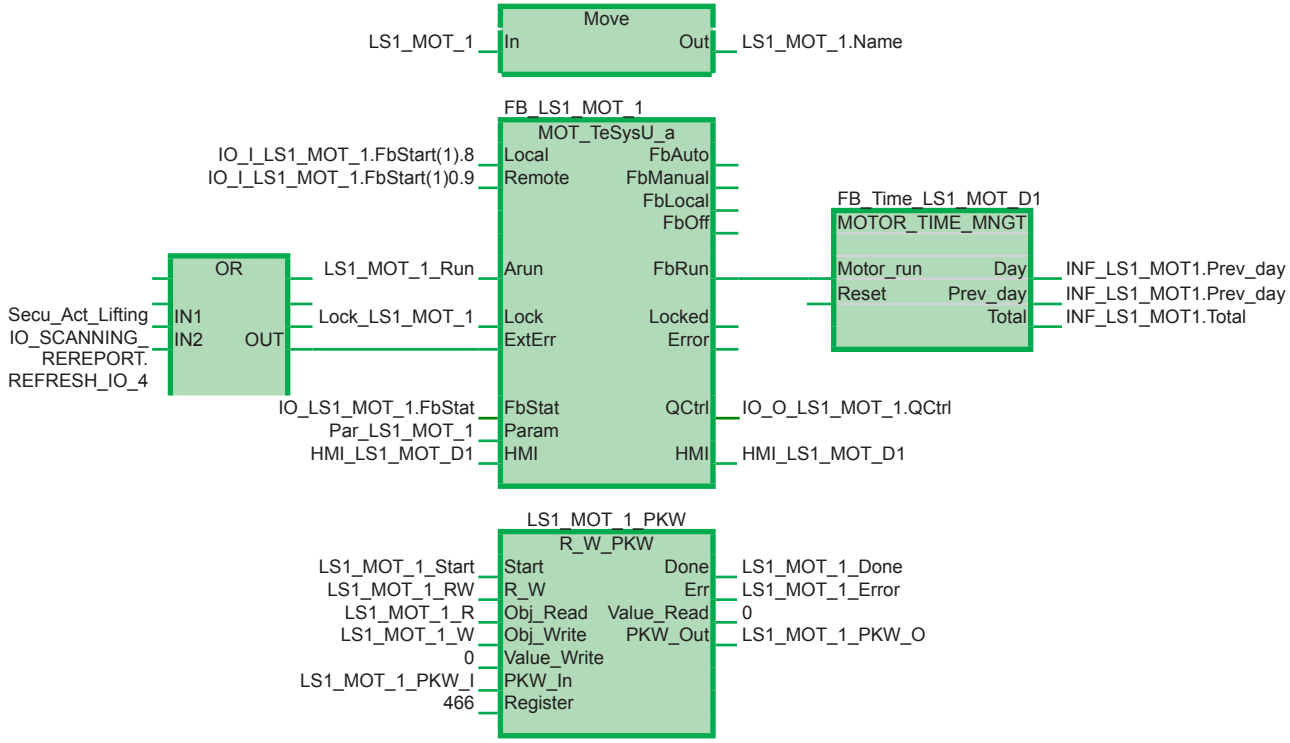
2-Example of starter type ATV61, one block to display the name on the HMI, one block for starter management and an optional block for time management



Implementation

Unity program (cont'd)

3-Example of starter type TeSys U with advanced control unit,, one block for the naming, one block for starter management and an optional block for time management There is an additional block allows Read or Write all TeSys_U parameters and data.



Following Tag naming rules defined in chapter 5.
LS1_MOT_1LS1 = Lifting Screening Unit 1MOT_1 = Motor Number 1

Operation

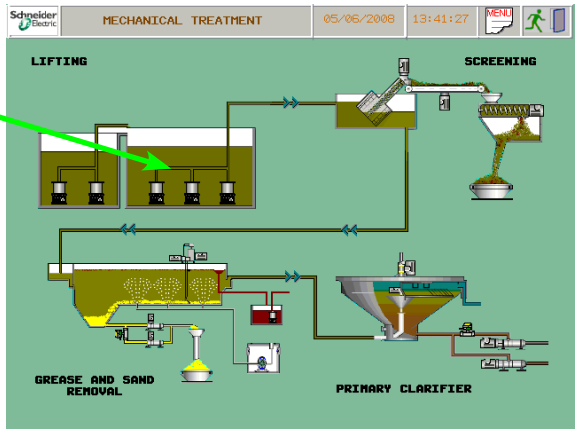
Introduction

The water application can be operated either from a Vijeo Citect SCADA for complete process monitoring or from a Magelis HMI more dedicated to local monitoring and maintenance purposes.
In Local mode, button panels can also be implemented to perform pump and motor management with no PLC control.

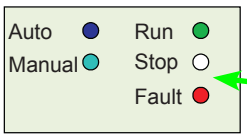
In the general process view, a functional unit can be selected to visualize the pump to be controlled.

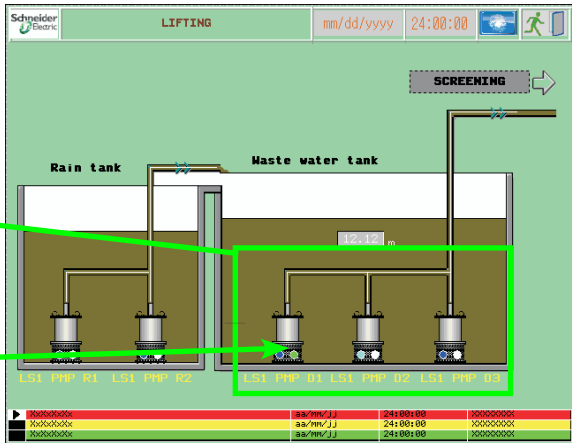
Control pump example

Click the functional unit to zoom in

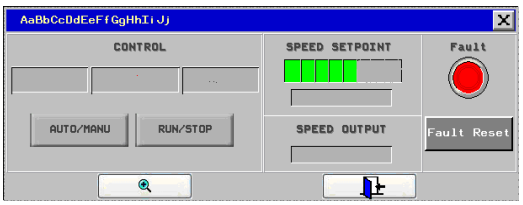


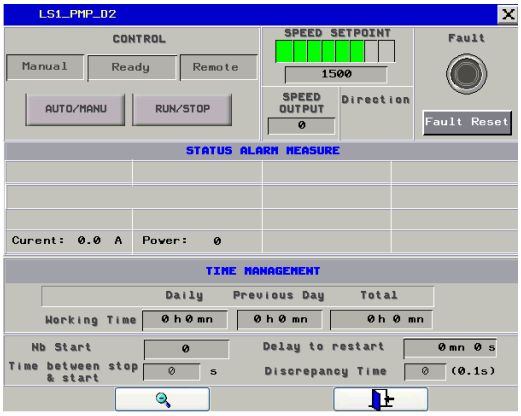
A summary report of the pump is provided by indicator lights





To perform a control operation on the pump, click the object (see the two levels below)

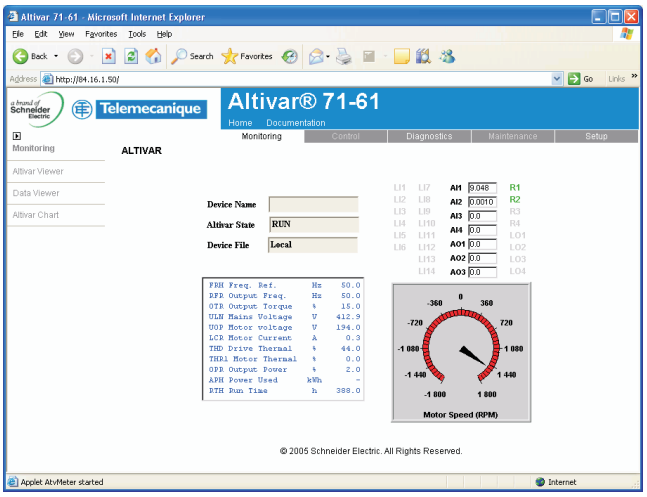




Operation

Control pump example

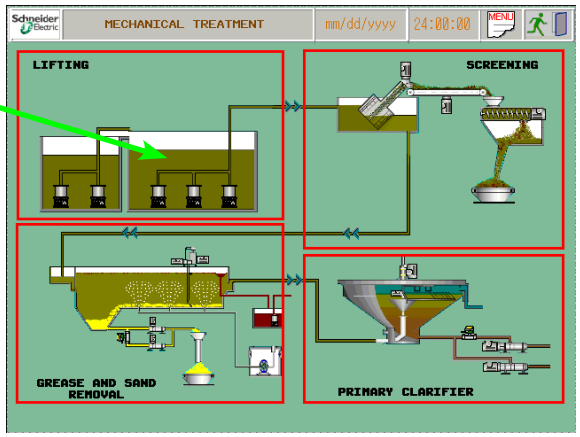
Using a web browser provides access to additional information.



Faulty pump example

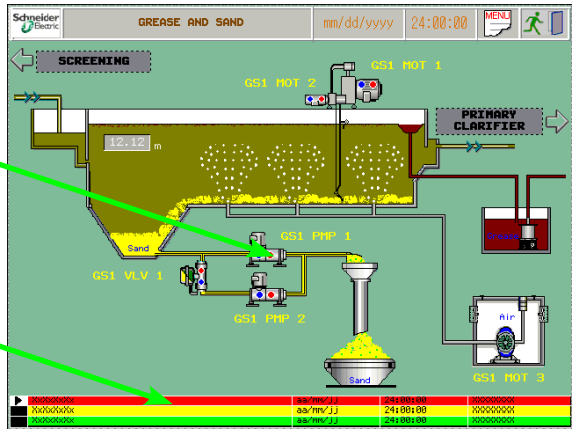
When a fault is triggered, an alarm message is sent and the information appears in the general view. Click the area to access the defective element. An alarm message is sent.

Clic on the functional unit to zoom in



Click on the pump to diagnose it

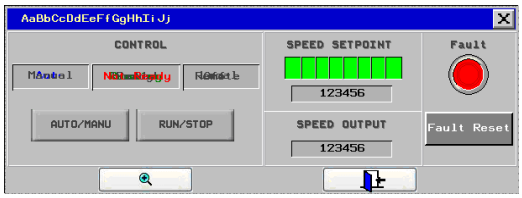
Alarm

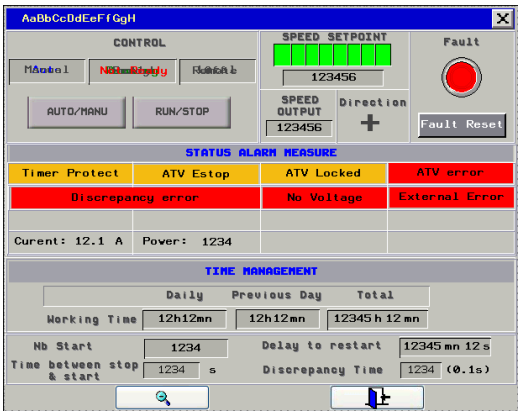


Operation

Faulty pump exampl(cont'd)

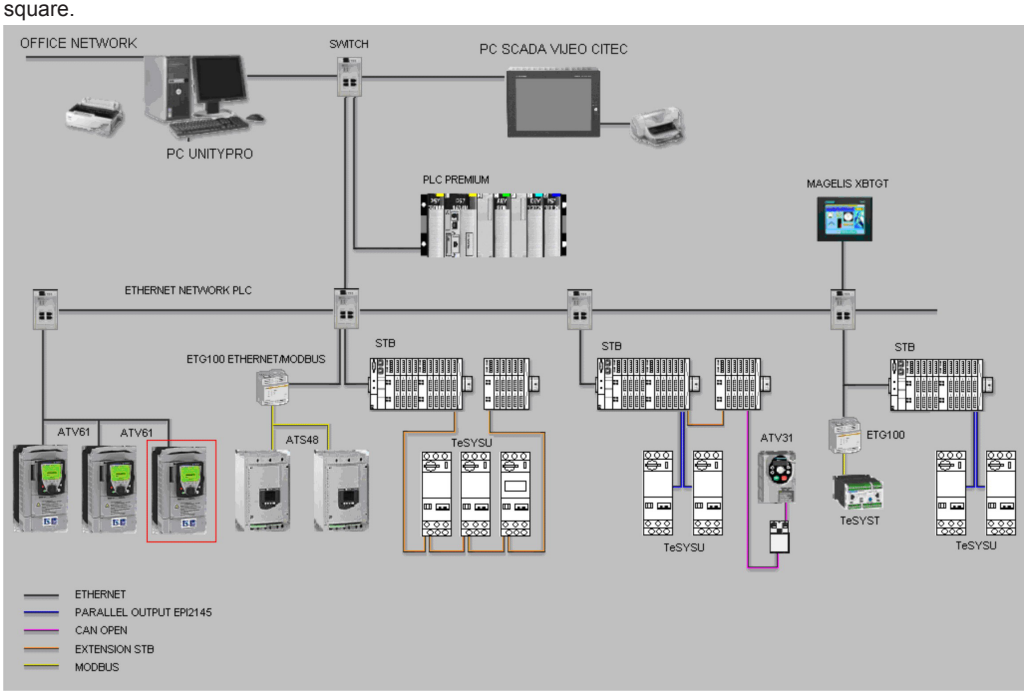
The two popup levels below provide access to detailed information about the pump.





Network faulty example

Following sceen capture represent Ethernet architecture, Faulty communication is warned by a flashing red square.



Components list

Part Number	Description
Control Room	
TSXP574634M V2.40	PLC Processor with Ethernet port
TCSESM083F23F0 SV1.03	ConneXium managed Switch 8 x 10/100Base-T ports
MPCKT22NAN00N-PP SV1.0	Magelis Compact iPC 12"
ABL8RPS24050	Phaseo power supply 230 VAC,120 W, 5A 24V DC
Lifting & Screening	
STBNIP2212 V2.14	TCP/IP communication module for STB
TCSESM083F23F0 SV1.03	ConneXium managed Switch 8 x 10/100Base-T ports
499NES25100 SV1.03	ConneXium Switch 5 x 10/100Base-T ports
TSXETG100 V2.50	Modbus/Ethernet bridge
LC1D09BD	Contactor up to 9 A, 24 V
LC1D18BD	Contactor up to 18 A, 24 V
GV2-L08	Motor circuit-breaker, 4A
GV2-L20	Motor circuit-breaker, 18 A
W3A3310 V2.1	ATV Communication Card for Ethernet
ATV61HO75N4 V1.4	Variable Speed Drive for ATV 61, 0.75 kW , 3~, 400V
ATS48D17Q V1.1	Soft Starter 7.5 kW, 3~, 400V
LUCA05BL	Unit Control TeSys U standard 0.25…15KW
LUCB05BL	Unit Control TeSys U advanced 0.25…15KW
LUCM05BL	Unit Control TeSys U multi-functions 0.25…15KW
(MG)28120	Master switch NSC100N TM16D
ABL8RPS24050	Phaseo power supply 230 VAC,120 W, 5A 24V DC
XPSAF5121	Safety relay Preventa
LC1D25BD	Contactor up to 25 A, 24 V
Grit & Grease Removal	
STBNIP2212	TCP/IP communication module for STB
499NES25100	ConneXium Switch 5 x 10/100Base-T ports
STBEPI2145	Adantys STB Tesys U Extension Module
LUFC00	Interface EPI2145
LUCA05BL	Unit Control TeSys U standard 0.25…15KW
LUCB05BL	Unit Control TeSys U advanced 0.25…15KW
LC1D09BD	Contactor up to 9 A, 24 V
GV2-L08	Motor circuit-breaker, 4A
ATV31HO37M2 V1.7	Variable Speed Drive for 0.37kW 230V - 3~
LTMR08MBD	Unit Control TeSys T, Modbus, 0.4 to 8A, 24VDC
(MG)28120	Master switch NSC100N TM16D
ABL8RPS24050	Phaseo power supply 230 VAC,120 W, 5A 24V DC
XPSAF5121	Safety relay Preventa
LC1D25BD	Contactor up to 25 A, 24 V
Primary Clarifier	
STBNIP2212 V2.14	Interface Ethernet pour STB
STB EPI2145	Adantys STB Tesys U Extension Module
LUFC00	Interface EPI2145
XBTGT4230 SV1.1	Graphic HMI 7.5 ''
LUCA05BL	Unit Control TeSys U standard 0.25…15KW
LC1D09BD	Contactor up to 9 A, 24 V operation,GL
GV2-L10	Motor circuit-breaker, 6.3 A
ABL8RPS24050	Phaseo power supply 230 V,120 W, 5A, 24V DC
XPSAF5121	Safety relay Preventa
LC1D25BD	Contactor up to 25 A, 24 V
Software	
Unity Pro 3.1 XLS	Controller Configurating & Programming Software
Vijeo Citect 7.0	SCADA Software
Vijeo Designer 4.6	HMI Software
Advantys Configuration Tool 2.5.0.1	Advantys STB Configuration Software
Power Suite 2.5	Altivar / TeSys Configuration Software
OFS 3.31	OPC Server

8-Glossary

Glossary / Acronym	Description	Page
Cicode Function	Cicode is a Vijeo Citect programming language designed especially for plant monitoring and control applications. Using Cicode, you have access to all real-time data in the Vijeo Citect project and all Vijeo Citect facilities. Cicode can also be used as interface to various resources on computer.	53
DDT	DDT (Derived Data Type) is a set of elements of the same type (ARRAY) or of various types (Structure).	31-37
DFB	A DFB (Derived Function Block) is a user function block that has been customized to take the specific nature of your project into consideration. It can be stored in the User-Defined Library.	27-28
Distributed Automation Island	Or called distributed I/O, is a distributed I/O unit installed in the field. The communication between I/O islands and CPU is via fieldbuses on which IO islands are defined as nodes.	14
DOL	Direct On Line is the simplest motor control mode which only uses circuit breaker and contactor to control the start/stop of motors.	8-14
Embedded diagnostic service (web technology)	Embedded diagnostic services are functions provided by some intelligent Ethernet devices, which can perform hardware error diagnosis and logic error diagnosis. Embedded web pages or other tools are used to indicate these errors.	15
Genie and Super Genie	Genies are combinations of several objects in Vijeo Citect Symbol Library. It can be used as single object and the elements of the genie are then configured collectively. Super Genies are dynamic pages (usually pop-ups), to which you can pass information when the page displays in the runtime system.	27-31 51
Heatsink	A heatsink is an environment or object that absorbs and dissipates heat from another object using thermal contact (either direct or radiant).	9
HMI	Human-Machine Interface	15;19
I/O Scanning	IO scanning is a scheme that periodically read or write to/from remote inputs/ouputs on the Ethernet network, without requiring any specific programming.	34-36
iMCC	Intelligent Motor Control Centers	2
IP	Internet Protocol	34
multi-pump card	A multi-pump card is an accessory for Altivar 61 which can adapts the drive to pump application. Multiple pumps can be controlled by Altivar 61 with different operating mode.	2
OFC Server	Object-Linking and Embedding (OLE) for Process Control, is designed to bridge Windows based applications and process control hardware and software applications that permits a consistent method of accessing field data from plant floor devices.	
PLC	A programmable logic controller is a digital computer used for automation of industrial processes. Unlike general-purpose computers, the PLC is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed or non-volatile memory.	3-7
Popup Page	A popup page is a dynamic page to that you can use to pass information when the page displays in the runtime system. The same page can be reused with different sets of tags.	27;53
Power Removal (PWR)	"Power Removal" is an embedded safety function for ATV61/71 which prohibits unintended equipment operation. When Power Removal function is triggered, the motor no longer produces torque.	21
Redundant	Redundant architecture is used to raise the availability of application. Redundant architecture is composed of primary unit and standby unit. When an error induces malfunction of primary unit, standby unit will replace primary unit to ensure the performance of certain functions.	61
SCADA	A Supervisory Control and Data Acquisition is a system that sends commands to a real-time control system to control a process that is external to the SCADA system. This implies that the system coordinates, but does not control processes in real time.	2; 19 31
Soft Starter	A motor soft starter is a device used with AC electric motors to temporarily reduce the load and torque in the powertrain of the motor during startup.	7-14 23
Standalone SCADA	A standalone SCADA is a supervisory architecture with both Server and Client Functions integrated within one operator station.	62
STG	A System Technical Guide is provided with each "How can I" project. Base on application lifecycle, STG will provide a detail guide on each phase to meet your control requirements.	2-3
TCP	Transmission Control Protocol	14;26
Unlocated variables	An unlocated Variable is a variable for which it is impossible to know its position in the PLC memory. A variable which has no address assigned is said to be unlocated.	62
VSD	A Variable Speed Drive (VSD) is a system for controlling the rotational speed of an alternating current (AC) electric motor by controlling the frequency of the electrical power supplied to the motor.	8-9 14;18
XVM file	XVM is a XML format file generated by Unity Pro. When you export variables, all the unprotected information will be stored in a XVM file which is compatible to OFS.	26;39 40-42

Schneider Electric Industries SAS

89, bd Franklin Roosevelt
F-92505 Rueil-Malmaison
FRANCE
Tél. : +33 (0) 1 41 29 70 00
www.schneider-electric.com

Due to evolution of standards and equipment, characteristics indicated in texts and images in this document are binding only after confirmation by our departments.
Publication: Schneider Electric